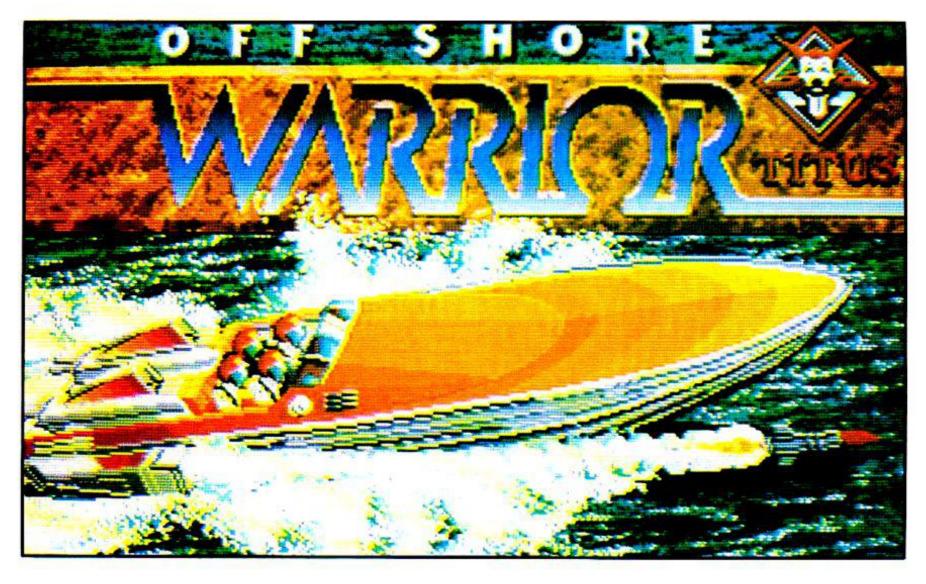
Commodore
COMPUTER
CLUB
66
L. 5.000
La rivista degli utenti di sistemi Commodore
Anno VIII - N. 66 - 25 Luglio/25 Agosto 1989
Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe
Centra l'estate
con l'Amiga
e il C64
Amiga, 1541, C128
i superquiz
Porte logiche,
ultime novità
Quanto costa
la tua vacanza
C 128,
a tutto
Assembly
systems

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Laur a & Miria Colombo, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Domenico Pavone, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma, Franco Rodella, Stefano Simonelli
Grafica: Arturo Ciaglia
Direzione, pubblicità: via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310 - **Redazione:** Tel. 02/5249211
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Angelo Ricupero - Via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Arretrati e software: Milano, V.le Famagosta, 75 - tel. 02/8467348 - Sig. ra Lucia Dominoni (Il servizio è operativo nelle ore pomeridiane. Nelle altre ore rivolgersi allo 02/55500310)
Tariffe: prezzo per copia L. 5.000. Abbonamento annuo (11 fascicoli) L. 50.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 90.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - Jonathan - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - TuttoGatto Videoteca - VR Videoregistrare

UN FASCICOLO TUTTO ESTIVO

Durante le vacanze è necessario riposarsi; ma ciò non vuol dire abbandonarsi all'ozio

Come nostra consuetudine, consolidata negli anni, il fascicolo estivo contiene argomenti di tutto riposo e di disimpegno (quasi) totale.

Tutti noi, infatti, desideriamo riposarci dopo un anno che, chissà poi perchè, risulta sempre essere stato molto più faticoso dell'anno precedente.

Se i primi giorni di vacanza, però, li passiamo ad ubriacarci della libertà riconquistata, verso il quarto giorno inizia a manifestarsi quel disagio che, in un modo o in un altro, rischia di trasformarsi ben presto in noia.

Diamine, se avessimo un computer qui, sotto l'ombrellone, potremmo verificare se quella routine... potremmo modificare quell'altro programma... potremmo aggiungere una certa opzione in modo che...

Vorrei, ma non posso.

Capita così che tutte le volte che abbiamo tanto tempo a disposizione, puntualmente manca il calcolatore per mettere a punto quel listato che, nel periodo normale di lavoro, siamo sempre stati costretti ad accantonare.

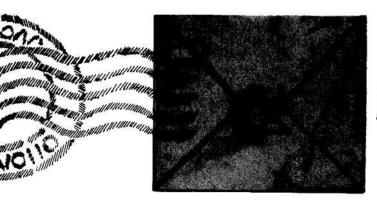
Bene, non dico che ora vi forniamo un computer a batterie allegato al fascicolo che avete in mano, però abbiamo pensato a tener sveglia ed attiva la vostra mente; basta che lo vogliate.

Portate, quindi, *Commodore Computer Club* sulla spiaggia e leggetelo con attenzione: vi sono diversi quiz, "sfide" di vario tipo, suggerimenti per il software da sviluppare non appena tornerete a casa.

L'importante è non abbandonare questo meraviglioso hobby, sia che possediate il C/128, il C/16 o l'Amiga (non preoccupatevi, il C/64 resiste ancora!).

A settembre, ve lo assicuro, torneremo alla carica con nuovi ed affascinanti argomenti, da sviluppare nel lungo periodo invernale. Ma ora basta con i pensieri foschi, relativi a scuola o ufficio, e correte a tuffarvi in mare.

E, dato che siete in acqua, provate a determinare un algoritmo che, tenendo conto della forza del vento, faccia muovere uno sprite - barca lungo lo schermo mentre, con un joystick, un omino in windsurf...

Alessandro de Simone

la vostra posta

CAMBIAR REGISTRO

□ **E' possibile modificare o aggiungere righe di programma mentre questo è in esecuzione?**
(Mario Sammartino - Capua)

• L'operazione di cancellazione, modifica o aggiunta di righe Basic implica, da parte del computer, l'esecuzione di un numero notevole di istruzioni.

Bisogna pensare, infatti, che le variabili numeriche e stringa (compresi vettori e matrici) sono posizionate automaticamente subito dopo la fine del Basic, tranne nel caso del C/128, che le alloca nell'altro banco di memoria Ram a disposizione.

E' quindi intuitivo che l'immissione di una nuova riga Basic costringe il computer a *shiftare* l'intera zona delle variabili di tanti byte quanti ne rappresenta la nuova riga.

Di solito, inoltre, l'immisione (o la semplice modifica) anche di una sola istruzione Basic può alterare l'intero "percorso" logico del programma. Ne consegue che se si interrompe una certa elaborazione, già impostata in un determinato modo, l'aggiunta, la modifica o la cancellazione di una riga potrebbe creare notevole pericolo di conflittualità nell'algoritmo di esecuzione.

Per evitare tale inconveniente i progettisti dei linguaggi interpreti hanno impedito la prosecuzione di una elaborazione (effettuabile con il famoso *Cont*, in caso di interruzione dello stesso) nel caso in cui vengano apportate modifiche al programma.

Naturalmente è possibile ovviare al piccolo "inconveniente" agendo, mediante *Poke*, all'interno della stessa area del programma; ma spieghiamoci meglio con un esempio.

Digita il seguente programma:

```
100 rem eccomi qua
105 x=2053: rem byte che contiene il primo token
110 for i =2053 to 2060
120 print peek(i);: next
130 if peek(x)=143 then poke x, 153: poke 2054, 34: goto 150
140 poke x, 143: poke 2054, 32
150 print: print: goto 100
```

La locazione 2053 (stiamo parlando di un C/64) contiene il *Token* (cioè il codice relativo) della prima istruzione della prima riga di un qualsiasi programma Basic. Nel nostro caso, all'istruzione *Rem* è associato il valore 143.

Poichè tale codice, come tutti gli altri, sono allocati in Ram, è possibile modificare, con una semplice Poke, il suo valore e, di conseguenza, il Token della prima istruzione.

Il programma non fa altro che, alternativamente, trasformare Rem in Print (codice 153) e viceversa. Inoltre, nella locazione successiva a 2053, viene alternativamente depositato il carattere di spazio (32) o apici (") di codice 34.

In questo modo si possono apportare modifiche anche sostanziali ad un programma *durante* la sua esecuzione impedendo, alla routine di controllo, di intervenire con il messaggio "Can't continue error" nel caso avessimo fermato l'esecuzione, modificato Rem in Print, e ripreso l'esecuzione.

Studiando opportunamente i vari listati specifici, ed individuando con la massima precisione le locazioni che contengono i vari token, è possibile intervenire in modo più massiccio su qualsiasi programma.

Dirò di più: molti esperti programmatori approfittano della tecnica descritta per allocare, all'interno di righe Basic, mini-routine in linguaggio macchina o messaggi di copyright da attivare in determinati momenti.

QUI PC, RISPONDETE!

□ **Sarei disposto ad acquistare un XT compatibile a patto di poter "riversare", su programmi professionali Ms-Dos, tutti i file che, negli anni precedenti, ho creato con il mio C/64 servendomi di programmi tipo Easy Script, Calc Result, Superbase e così via. E' possibile un'operazione del genere?**
(Fabio Ruffini - Brescia)

• I file generati da Easy Script sono dei banalissimi file Ascii che, mediante opportuno software e interfaccia (tra cui, lo ricordo, quelli descritti minuziosamente sul N. 49 di C.C.C.), è possibile trasferire e manipolare con un qualsiasi PC compatibile corredato da qualsiasi Word Processor esistente.

E' bene dilungarsi, però, cui codici speciali eventualmente presenti nei file; mi riferisco, ad esempio, al carattere di asterisco in campo inverso (generato da F3 con Easy Script) che manipola il formato del testo (margini, giustificazione, numero di righe, intestazioni e così via). Questi codici non possono essere riconosciuti da altri w/p per il semplice motivo che, almeno

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

Per accelerare il servizio, ricordate di indicare sempre la data di spedizione dal momento che questa costituisce diritto di precedenza.

A tutt'oggi (con la pubblicazione del presente fascicolo) risulta evasa l'intera corrispondenza pervenuta in Redazione entro la fine del mese appena trascorso.

Non ci è possibile dare consigli su acquisti da effettuare nè esprimere pareri estranei alla logica della correttezza professionale e commerciale.

per quanto mi risulta, nessuno ha mai pensato ad effettuare le conversioni necessarie. Non dovrebbe, però, esser difficile, ad esempio, studiare i codici di Easy Script e di Word Star (tanto per citare il più diffuso w/p che gira in ambiente PC) e stendere il corrispondente programma di conversione in modo tale da ottenere, con un PC, lo stesso, identico risultato che si otteneva con Easy Script e C/64.

Se, però, la conversione risulta relativamente semplice per i file generati da un Word processor, la faccenda diventa davvero ardua cercando di caricare, per esempio con *Quattro* (uno spreadsheet piuttosto noto, in ambiente Ms-Dos, per la sua potenza), un file generato dal "collega" Calc Result, altrettanto noto sul pianeta C/64.

I codici usati, l'algoritmo seguito, e le varie elaborazioni impostate nei due programmi sono strepitosamente diversi e non penso che vi sia, al mondo, un essere raziocinante, ancora vivo, che abbia affrontato il problema della convertibilità dei file.

Forse per un Data Base la conversione può risultare più semplice (e rimane più complessa della conversione di un file Ascii di un w/p) ma, in ogni caso, è pur sempre necessario un programma *specifico* di conversione.

Segui il mio consiglio: al negoziante che ti assicura la fattibilità della conversione, prometti di acquistare il computer che ti propone solo *dopo* che ti dimostrerà la conversione stessa. Digli, da parte mia, che pubblicherò *gratis* il suo nominativo (completo di telefono) se, in effetti, è in grado di consegnarti, insieme al computer, i programmi di conversione che ti interessano e che farebbero felici numerosi lettori, disposti a sborsare parecchi quattrini per ottenere i programmi in questione.

In caso contrario, rivolgiti a qualche rivenditore più serio, in grado di promettere di meno, ma di mantenerlo...

PROTESTE

☐ **Vorrei sollevare alcune osservazioni sul modo in cui conducete la rivista Commodore Computer Club...**
(Simone Franceschi - Maresca)

• *"Non dedicate sufficiente spazio al C/128* -prosegue il nostro lettore- *le recensioni, quando ci sono, sono "oscene", il prezzo di copertina è troppo elevato in confronto a ciò che offrite e molti argomenti, inoltre, sono affrontati in modo decisamente infantile. Purtroppo* -e così conclude la lettera- *sono costretto a comprarvi perchè non ho altro mezzo con cui imparare a programmare".*

L'ultima frase, per giunta conclusiva delle proteste, è certamente (e, forse, involontariamente) illuminante.

Dunque: le altre riviste costano di meno, pubblicano foto di computer invece di ragazzotte più o meno vestite (per non parlar dei topini), offrono argomenti spiegati in modo più accurato, però... solo con C.C.C. è possibile imparare a programmare.

Non si dispiaccia il nostro lettore per quello che sto per dire e che, comunque, riassumo brevemente.

La nostra rivista offre un "servizio" che altre riviste non offrono: la didattica che, scusate l'immodestia e la presunzione, riteniamo di affrontare in modo più che "decente".

Noi non siamo destinati a chi vuol far credere agli altri (ma anche, purtroppo, a se stessi) di intendersi di computer mentre, al contrario, non è affatto vero. Non ci interessa, per analogia, la gente che compra periodici finanziari per far capire che se ne intendono di Alta Finanza mentre non sanno la differnza tra un BOT e un CCT.

Ditemi voi che diavolo può capire un utente quando, su una rivista (magari patinata e di 300 pagine) trova articoli sulla gestione, in rete, del quarto bit appartenente al settimo byte dell'intestazione di un file manipolato in interrupt; e, soprattutto, a che può *servire* un'informazione del genere.

L'invito che rivolgiamo ai nostri lettori (che, *speriamo*, comprano altre riviste) è di svolgere la ricerca seguente:

1- prendete gli ultimi tre numeri di C.C.C. e gli ultimi tre numeri di una qualsiasi pubblicazione concorrente.

2- sfogliate le pagine dei sei fascicoli e (mi raccomando, siate sinceri con voi stessi!) annotate il numero di pagine realmente *comprensibili*.

3- come al punto 2, individuate le pagine (listati, articoli e così via) che ritenete realmente *utili*.

4- tirate le conclusioni.

ACCESSORI POCO CITATI

☐ **Mi sono procurato un accessorio, per il mio C/64, descritto tempo fa sulla vostra rivista. In seguito, rispondendo ad un lettore, avete affermato che, a causa del mancato successo di detto accessorio, non ne avreste par-**

OPINIONI

La rubrica "La voce dei lettori" è nata per dare spazio (e sfogo...) ai lettori che, pur non avendo nulla di particolare da chiedere, intendono far sentire la propria voce, chiedere solidarietà ad altri utenti o, semplicemente, riportare osservazioni ritenute utili o importanti. Per questioni di spazio le lettere sono spesso condensate; vengono poi "censurate" le osservazioni ed i termini troppo "forti" che possano arrecare offesa ai lettori, agli inserzionisti e... a noi stessi. In calce viene riportata la sola iniziale del cognome del lettore (salvo casi particolari), per fare in modo che questi possa sentirsi ancor più libero di esprimere la propria opinione. Coloro che desiderano il minor numero di "tagli" possibile alle loro missive sono pregati di inviarle su DISCO, usando il word processor Easy Script.

lato. E' questo il vostro modo di far didattica?
(Pasquale L.M. - Agrigento)

• Ho riletto l'articolo citato dal nostro lettore (che, tra l'altro, ho scritto proprio io) e non capisco la lamentela.

Tutti (o quasi) gli articoli di qualunque periodico contengono informazioni *esplicite* ed informazioni nascoste *tra le righe*.

La legge italiana, in materia di giornalismo, impedisce (pur se *tra le righe* anch'essa...) di esprimere commenti troppo espliciti: si può incorrere facilmente nel raeato di diffamazione.

Pertanto, quando diciamo che "questo accessorio sarà apprezzato quasi esclusivamente dagli esperti di linguaggio macchina... peccato che non vi siano programmi specifici... abbiamo rilevato un bug imbarazzante... le istruzioni sono poche, poco chiare e solo in inglese" chiunque dovrebbe capire che *tra le righe* (ma, in questo caso, piuttosto esplicitamente), cerchiamo di inviare un messaggio di *Attenzione!* se non di vero e proprio *Allarme!*

E' vero che un qualsiasi scritto, inevitabilmente, ha molte probabilità di esser condizionato dalle opinioni personali di chi lo scrive: ma in alcuni casi sarebbe opportuno riflettere un paio di volte prima di "forzare" il destino.

Il fatto, poi, che detto accessorio sia stato un fallimento commerciale, dimostra che il messaggio inviato tra le righe del famoso articolo era basato su valide considerazioni. Se non bastasse, la tua è la prima (ed unica) lettera che richiede argomenti specifici.

LA VOCE DEI LETTORI

VIVA IL SIMON'S BASIC

Commodore Computer Club si ostina a fare i programmi in linguaggio Gw-Basic. Ad esempio, sul N. 63 c'era un programma di musica intitolato "Come Pavarotti con il C/64", ma anche altri programmi pubblicati in precedenza sono in linguaggio Gw-Basic.

Sembra che il famoso *Simon's Basic* sia scomparso o dimenticato, come buttato nel cestino. Io ho provato a tradurre, con difficoltà, i programmi per il Simon's Basic, ma con risultati deludenti. Non dimenticate che nel mondo dell'informatica esiste anche il Simon's Basic.
(Luca Manganelli - Grosseto)

COMANDI QUASI ASSENTI

□ Il mio C/128, digitando Quit e Off, non risponde con un Syntax error (come dovrebbe) ma con un inspiegabile Unimplemented command error. Che cosa vuol dire?
(Federico Alpi - Roma)

• Tutte le volte che si impartisce un comando, il computer (o meglio, il suo interprete Basic) confronta, quanto digitato, con le parole chiave contenute nelle sue Rom; se il comando effettivamente esiste, viene attivata la corrispondente routine l.m. che si incarica di svolgere la funzione desiderata.

In alcuni casi, ed è quello del C/128, i progettisti decidono, in un primo tempo, di inserire particolari forme sintattiche che però, in un secondo momento, decidono di eliminare o di sostituire con altri comandi (forse *Quit* è stato sostituito da *Go 64?*). Per evitare di riscrivere le Rom, preferiscono lasciare al loro posto i *Token* dei vari comandi, rendendoli, di fatto, inutilizzabili. In pratica il comando viene, sì, "riconosciuto", ma questo attiva, in ogni caso, la routine di errore segnalata dal nostro lettore.

E' però probabile che, all'interno delle Rom del C/128, siano ancora presenti le routine "escluse" e che, mediante una Sys, potrebbero egualmente essere attivate. Chissà, però, se ci sono e, soprattutto, *dove* sono.

MOUSE E SPRITE

□ Quali sono le locazioni di memoria in cui leggere le coordinate del mouse e come gestire gli sprite con il vostro Gw-Basic Emulator 64?
(Rosario De Chiara - Salerno)

• Il mouse della Commodore, da collegare al C/64, non è altro che un joystick un po' speciale. Spostandolo a destra, infatti, provoca lo stesso effetto del continuo spostamento (e successivo "rilascio") dell'assicella del joy nella stessa direzione, finchè non si arresta il movimento del "topo".

Per determinare la direzione del mouse è quindi sufficiente esaminare il contenuto di due locazioni, o meglio i singoli bit dei byte che li contengono. Le locazioni interessate sono la 56320 (per la porta 2) e 56321 (per la porta 1) ma quest'ultima può presentare problemi durante il funzionamento di programmi Basic.

Un programma per verificare quanto detto può essere il seguente...

```
100 Print Peek (56320): Goto 100
```

...valido sia per il mouse sia per il joystick collegati in porta 2. Mentre "gira", prova a spostare, lentamente, il mouse in tutte le direzioni.

Per ciò che riguarda la gestione degli sprite con il Gw-Basic Emulator non posso fare altro che ripetere quanto già detto più di una volta: è possibile manovrare sprite con le solite Poke con cui si manovrano avendo a disposizione il C/64 nudo e crudo; il Gw-Basic, infatti, non contiene istruzioni particolari per gli sprite che, nei "veri" Ms-Dos, non esistono nemmeno.

STRANE RICHIESTE

□ Ho adattato, con il vostro Gw-Basic Emulator per C/64, un programma che, in origine, era destinato per l'IBM. Purtroppo non funziona come dovrebbe e vi invio il dischetto per individuare l'errore. *(Fabio R.)*

Il programma che ho scritto (e la cui copia vi invio su carta) gestisce una biblioteca su nastro. Vorrei adattarla per disco, dal momento che ho acquistato un drive. *(Dimitri)*

Il programma "The Bridge" (C.C.C. N. 60) che ho copiato dalla vostra rivista, emette un Syntax error la cui natura

LA VOCE DEI LETTORI

VIVA IL PLUS/4

Finora, per non parlare del +4, avete sempre cercato scuse del tipo: *"se ne sono venduti pochi"* oppure: *"non ci sono in commercio programmi specifici"*. Non voglio dire che il ragionamento sia sbagliato, ma neanche tutto vero. Probabilmente in questi ultimi anni avranno venduto "qualche" Plus 4, soprattutto grazie all'inevitabile calo del prezzo.

Non tutti possono spendere cifre per soddisfare la loro sete di informatica. Tra questi, purtroppo, anch'io. Così ho acquistato un +4 (che costava quasi 200 mila lire in meno del C/64, due anni fa) ma vi assicuro che non ne sono pentito assolutamente. Anzi: non so, dei 64-isti, quanti conoscono almeno un po' di questo computer: di sicuro, se lo conoscessero meglio, non potrebbero che restare a bocca aperta per la sua facile gestione: vedi tasti programmabili, comando *Escape:* una bomba con finestre in qualsiasi parte del video premendo solo due tasti, per non parlare dei semplici e potenti comandi grafici e sonori, del monitor e dei programmi professionali incorporati. Questi ultimi, in verità, avrei dovuto citarli per primi, visto che sono una novità per i piccoli CMB e per i computer in generale. Programmi, questi, che il C/64 esegue non senza l'iniezione di una cartuccia o un dischetto e, poi, quando hai l'uno non puoi avere l'altro.

Mi piacerebbe che queste cose le potessero sapere i 64-isti irritati dalla sporadica comparsa di programmi per C/16 e +4 sulla vostra rivista, come se conoscessero a memoria tutto il ben di Dio che viene pubblicato ogni mese per il C/64! Penso che 3 o 4 pagine se le meriti anche "lui", sul vostro periodico, e su 100 pagine sono ancora poche. Tanto più che con il fratellino C/16, avendo come "alleato" il C/128, si raggiunge un numero di lettori più che meritevoli delle vostre attenzioni.

E non ditemi, per favore, che non potete sprecare energie per un computer così desolato!!! Anche se ho già deciso che appena totalizzo 13 al Totocalcio mi compro un mega computer a 126 bit con lettura di dischi ottici. Alias, esageriamo senza limitismo, per dirla alla Megasalvi.

Ho appreso, sempre da voi, che programmi specifici per +4 ce ne sono (anche se non facilmente reperibili): come avrei potuto cercarli se avete sempre cantato ai quattro venti che non ne esistevano? Con quale cognizione vi siete presi la libertà di dire una cosa simile? Mi sa che siete un tantino impreparati sull'argomento, lo prova anche il fatto riguardante la possibilità del Ted Chip di riprodurre la voce umana; avete ritrattato solo quando un attento lettore vi ha smentito con prove tangibili. Chissà quanti, come me, vorrebbero possedere un compilatore o un sintetizzatore vocale per Plus 4, quindi sarei grato se si spargesse la voce.

Ora che mi sono sfogato, vi auguro di continuare così. E poi, perchè non pubblicate quei programmi che vi vengono inviati? Avete sempre predicato che il mezzo migliore per aiutare la diffusione di un computer è lavorarci sodo; ma se poi evitate di pubblicare questi lavori, allora vi contraddite. Lo spazio per impaginare perfino fumetti però lo trovate: non dico che sia una brutta idea (perchè Primo Giovedini mi piace), però se esiste un problema di spazio, allora prima i programmi e la spiegazione che servono più allo scopo.
(Mauro C. - Mocasina)

non comprendo. Vi invio, su carta, il listato in modo che possiate aiutarmi.
(Alberto A.)

• Pubblichiamo volentieri le lettere dei nostri lettori per portarle come esempio di richieste che *non* possono esser prese in considerazione.

Un listato su carta, per quanto chiaro e "scorrevole", ci obbligherebbe a dedicare alcune ore di lavoro che, invece, utilizziamo con maggior efficienza per scrivere articoli, esaminare (ed esaudire) altre richieste e così via.

I listati allegati alle ultime due lettere, quindi, dopo aver loro dedicato il tempo che, democraticamente, dedichiamo a *ciascuna* lettera che perviene in Redazione, sono stati accantonati perchè non erano tali da far individuare facilmente l'errore. Posso solo ricordare a Dimitri che l'intero inserto del N. 39 era dedicato alla gestione dei file sequenziali su nastro e disco (compresi suggerimenti per eventuali conversioni).

Il primo lettore, invece, pur avendo esaudito la nostra richiesta di invio su disco, ha dimenticato una cosa importante e cioè in quale caso il programma non funziona. Purtroppo non siamo molto pratici di calcoli finanziari e, verificando i dati immessi con altri programmi in nostro possesso, il listato inviato sembra funzionare correttamente.

Morale della favola: cercate di essere chiari ed esaurienti nelle vostre richieste: a rimetterci, in caso contrario, siete (nostro malgrado) sempre e solo voi.

DA 1540 A 1541

☐ Ho acquistato un vecchio drive 1540 "trasformato" in 1541 per adattarlo al mio C/64. Purtroppo non sembra esser compatibile con alcuni velocizzatori. Quale ne è la causa?
(Firma illeggibile)

• Anch'io posseggo un glorioso 1540 trasformato in 1541 mediante la semplice sostituzione della Rom. A dire il vero lo utilizzo da anni e non ha mai presentato difficoltà di funzionamento nè con il Vic 20, nè con il C/16, nè con il C/64. Ho provato tuttavia, dopo aver letto la tua lettera, ad utilizzare software su disco e su cartridge, e non ho avuto problemi di sorta. Tieni presente, infine, che il mio ex 1540 è proprio vecchiotto, dal momento che lo acquistai con il Vic 20, appena fu commercializzato.

Ritengo, quindi, che il tuo drive presenti qualche "semplice" problema di allineamento che, non evidente in condizioni normali, si presenta invece in condizioni critiche, tra cui, appunto, i caricamenti turbo.

Amiga 2000

Amiga 2000 è il collaboratore ideale per preparare facilmente ed in pochissimo tempo al video testi e grafici e riportarli poi su carta, su lucidi o diapositive. È uno strumento creativo semplice e immediato per realizzare soggetti fantastici in due o tre dimensioni con 4.096 colori. Qualunque sia la tua professione, con Amiga 2000 hai a disposizione le meraviglie dell'ambiente Amiga DOS e del sistema MS-DOS, con i quali potrai elaborare i tuoi progetti con una grafica ineguagliabile. In più, Amiga 2000 è Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto da Commodore, che può così vantare la più lunga esperienza nel settore.

- MS-DOS e XENIX sono marchi registrati della Microsoft Inc.
- UNIX è un marchio registrato della AT&T.
- Commodore e Amiga sono marchi registrati della Commodore Inc.

Amiga 2000.
Per te
che vuoi
un amico
sulla
scrivania.

Amiga 2000.
Per te
che vuoi
lavorare
con un creativo
alla tua altezza.

Amiga 200
Per te e per
i tuoi progetti
che non hanno
bisogno solo
del sistema
MS-DOS.

C'è un Commodore p

LA VOCE DEI LETTORI

VIVA I PC COMPATIBILI

Ho acquistato un magnifico IBM PS/2, pur se continuo ad usare il fantastico C/64. A questo proposito ho notato che, con *Amigazzetta*, soddisfate le richieste degli utenti Amiga. Perchè non fate qualcosa anche per gli utenti dei PC compatibili?
Nel caso decidiate in modo affermativo, vi consiglio di divulgare programmi non solo su dischetti di formato 5.25, ma anche su mini dischi da 3.5 pollici: ormai questo formato sta conquistando il mondo dell'informatica.
(Luca B. Ravenna)

ATTESE COL REGISTRATORE

□ E' possibile evitare i cinque giri di attesa del registratore quando si apre un file di dati?

• Il computer, prima di inviare dati al registratore, trascrive un lungo "sibilo" che è possibile ascoltare inserendo una cassetta, contenente programmi, in qualsiasi mangianastri.

Tale sibilo sarà prezioso, nella successiva fase di lettura del programma o dei dati, per tener conto della velocità con cui opera il registratore. Potrebbe capitare, infatti, che il programma (o i dati) che il computer si accinge a leggere, siano stati registrati per mezzo di un registratore che, per vari motivi, possiede una velocità lievemente diversa da quella standard. Il computer, "ascoltando" eventuali anomalie nella riproduzione del fischio iniziale, si predispone automaticamente a ricevere i dati alla opportuna velocità.

In conclusione, i giri che il computer fa compiere al registratore prima di compiere la vera e propria registrazione, sono preziosi e indispensabili per consentirgli di adattarsi alla successiva rilettura.

IL PROGRAMMA C'E', MA NON SI VEDE

□ Nonostante il corretto allineamento della testina del registratore (posseggo addirittura un regolatore di Azimuth) alcuni programmi su nastro, dopo essere stati individuati, generano un "Load error" oppure un "Out of memory error". Come mai?

• E' probabile che siano programmi scritti per altri computer oppure registrati ricorrendo al Turbo Tape (utility per il Commodore 64 che velocizza la registrazione ed il caricamento dei programmi su nastro).

Nel primo caso, anche se riuscissi a caricarli, ti ritroveresti un programma molto probabilmente incompatibile col tuo computer, specie se scritto in linguaggio macchina (L.M.).

Nel secondo, invece, è sufficiente caricare, prima del programma che interessa, l'utility Turbo Tape; tieni presente, però, che sono in circolazione molte versioni di T/T, ognuna incompatibile con l'altra.

SIMULAZIONE DEL JOYSTICK

□ Non possiedo un joystick e desidero avere delucidazioni su come simulare, con la tastiera, le funzioni a cui assolve.

• Se intendi utilizzare, senza l'accessorio, i programmi in commercio, posso garantirti che non è assolutamente possibile effettuare le modifiche a meno di compiere tutte le operazioni che descrivo qui di seguito:

- Sproteggere il programma
- Disassemblarlo
- Individuare i segmenti di memoria preposti alla gestione del joystick
- Apportare le modifiche del caso.

Qualunque sia l'attività che svolgi, il tempo dedicato a risolvere il problema è decisamente più "costoso" (il tempo è denaro!) del prezzo dello stesso joystick (circa 20000 lirette).

Se, invece, vuoi apportare modifiche a programmi Basic per il Commodore 64 pubblicati sulle riviste, tieni presente che devi individuare tutte le righe di programma (che, in genere, contengono istruzioni del tipo Peek) relative alle locazioni 56321 (che controlla la Port 1) oppure 56320 (Porta 2). A seconda del valore contenuto in tali locazioni, è possibile conoscere in che modo è stata piegata l'assicella del joy:

125 SUD
126 NORD
123 OVEST
119 EST
111 FIRE

Ad esempio, un'istruzione del tipo...

```
200 x = peek (56320)
210 if x=125 then goto 1200
```

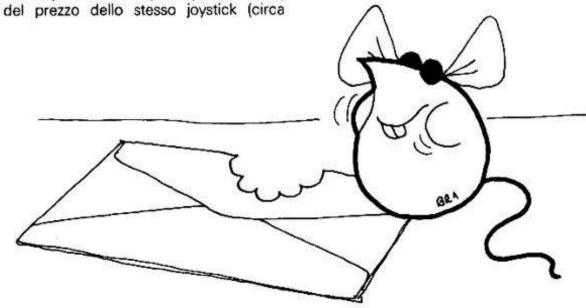
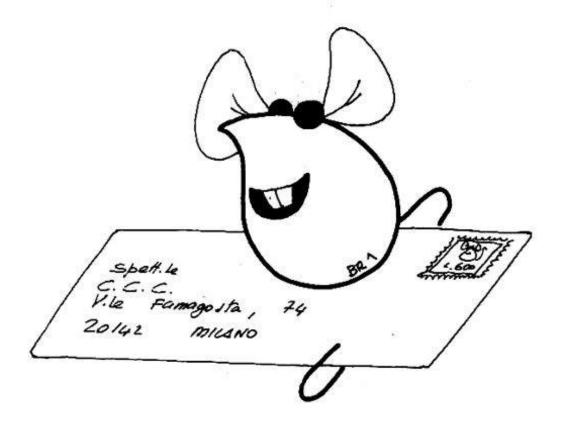
...devi convertirla in una del tipo:

```
200 get a$: if a$="" then 200
210 if a$="a" then goto 1200
```

Questo nell'ipotesi che si debba saltare alla riga 1200 nel caso si inclini il joystick a Sud oppure si prema il tasto "A".

LA MEMORIA DEL VIC II

□ Come è possibile spostare la memoria di schermo in modo da lasciare liberi i 1000 byte numerati da 1024 a 2023?
(da alcune telefonate)

• Lo schermo, che all'accensione del computer viene mostrato, è formato da 40 colonne e 25 righe, per un totale di (40 x 25 =) 1000 caratteri. Inoltre ogni carattere può avere un differente colore a scelta fra

16: in questo modo si può intuire che la memoria occorrente per la visualizzazione di 1000 caratteri, ognuno con un proprio colore, sia di 2000 byte.

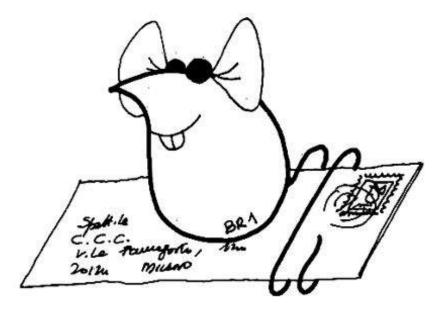
Il circuito integrato Vic II (6567) può gestire "solo" 16384 byte per volta (cioè 16K, valore dato da: 2 elevato alla 14-ma potenza) e, nel modo di funzionamento "normale", la memoria interessata è quella numerata da 0 a 16383. Questo è il primo dei quattro blocchi da 16K che il Commodore 64 possiede (16 x 4 = 64K).

Tale banco di memoria, che si può cambiare agendo opportunamente sulla locazione 56576, non viene, in genere, utilizzato per intero e, infatti, la memorizzazione di un programma Basic incomincia dalla locazione 2048 senza interferire con il regolare funzionamento del Vic II.

All'accensione, i 1000 byte occorrenti alla memoria di schermo vengono posizionati a partire da 1024 (la posizione può essere cambiata, sempre nell'ambito dei 16K scelti, agendo sulla locazione 53272) mentre quelli dedicati alla memoria del colore risiedono permanentemente nelle 1000 locazioni numerate da 55296 a 56295, e non possono essere modificate.

Spostando la memoria di schermo si deve prestare attenzione a non far collidere la memoria dedicata al Basic con quella, appunto, dedicata allo schermo.

A questo scopo è sufficiente modificare i puntatori 43 - 44 e 55 - 56 che indicano, rispettivamente, l'inizio e la fine dell'area di memoria riservata al programma Basic. E' inoltre necessario adeguare il puntatore locato in 648 che è usato dal sistema operativo per conoscere la locazione iniziale dello schermo.

Come intuitivo, la manipolazione di tutte le locazioni che abbiamo citato deve essere affrontata da utenti piuttosto esperti.

JOY E PADDLE

□ In un gioco in cui è previsto l'uso del joystick, è possibile usare le paddle (e viceversa)?

• Un joystick è formato da un gruppo di cinque interruttori che, a seconda della posizione dell'assicella, vengono chiusi oppure aperti e, in tal modo, il computer "riconosce" lo spostamento impresso.

Il joystick, pertanto, è uno strumento del tipo "tutto o niente" nel senso che un interruttore può esser solo aperto oppure chiuso, e non sono possibili "stati" intermedi.

Le paddle, al contrario, sono resistenze elettriche variabili che, collegate a porte analogiche, comunicano al computer una delle 256 possibili (almeno in teoria) posizioni intermedie della manopola.

Pertanto non è possibile la sostituzione accennata, sia perchè il joy è digitale e le paddle sono analogiche, sia perchè le paddle sono due e il joy possiede al suo interno ben cinque interruttori, sia perchè, infine, i pin di collegamento presenti nel connettore sono diversi nei due casi.

MANCANZA DI "PRESS PLAY"

□ Da alcuni mesi il mio computer, al comando Load e Save, non risponde più col solito messaggio "Press..." ma esegue immediatamente il programma, pur funzionando benissimo. Come posso fare per evitare questo difetto che è davvero seccante in programmi che richiedono esplicitamente la pressione del tasto Stop del registratore?

• Se il computer funziona egualmente nelle varie fasi (registrazione, verifica, caricamento) il difetto è da attribuirsi ad un corto circuito interpretato, dal computer, come un costante abbassamento di un tasto del registratore.

Tale corto circuito può essersi verificato sul computer (caso raro) oppure sul registratore (più frequente).

Per sapere quale dei due casi è il tuo, stacca il registratore e digita "Load". Se lo schermo "scompare" subito vuol dire che in corto è sicuramente la pista del calcolatore. Se, invece, compare la scritta "Press..." vuol dire che l'inconveniente è da ricercarsi nel registratore.

In entrambi i casi è necessaria un po' di esperienza per effettuare le necessarie riparazioni.

risposte rapide

FORMATO IFF
(Leandro Dardini - Firenze)

Il programma per Amiga, che affermi di aver scritto, potrebbe essere interessante a patto che abbia un effettivo contenuto didattico. Forse è meglio che telefoni per concordarne la stesura.

PUNTEGGIO ALTO
(Willy Polesh)

Non conosco il videogioco che indichi nella lettera e non posso quindi valutare se il punteggio da te raggiunto sia realmente alto come ritieni.

LEGGERE LE ISTRUZIONI
(Vincenzo Gulino - Palermo)

Le locazioni da manipolare per cambiar colore agli sprite, e per ingrandirli, sono spiegate con sufficiente chiarezza sul manuale di istruzioni del tuo C/64. Altre particolarità sulla gestione degli sprite sono riportate, di tanto in tanto, anche sulla nostra stessa rivista, che ti consiglio di leggere sempre con la massima attenzione.

CORSO DI L.M.
(Antonio Intini - Noci)

Dal N. 51 (marzo 1988) abbiamo trattato in modo sistematico il linguaggio macchina per il C/64 e, di tanto in tanto, anche le differenze esistenti con il C/128. Ti consiglio di procurarti tutti gli arretrati che mancano alla tua collezione per entrare in possesso di un "corso" sul linguaggio macchina certamente semplice.

NIENTE ERRORI
(Stefano Pederzani - Bologna)

Non vi sono errori nel listato da te citato; se il programma non funziona il motivo è da addebitare alla errata trascrizione del programma dalla rivista. Se ci invii il programma su disco possiamo aiutarti a rintracciarlo.

COMMODORE CAMBIA SEDE

La Commodore Italiana ha cambiato sede, e a partire dal 22 maggio di quest'anno, il nuovo indirizzo risulta essere:

Viale Fulvio Testi, 280
20126 MILANO
Tel. 02/66.123.1
Fax. 02/66.101.261
Telex 32.34.96

E' possibile raggiungere agevolmente la nuova sede anche con i mezzi pubblici:

Autobus: linea 44 da MM2 Crescenzago o Cascina Gobba. Linea 44 da MM1 Precotto.
Tram: Linea 2 da Stazione Centrale.

TESTI, LIBRI E VOLUMI

In lingua inglese, ma anche in italiano, è disponibile una vasta letteratura su Amiga; vediamo alcuni titoli

Gli Amici di Amiga

Parlando di Amiga, e dei libri ad esso dedicati, si incontra spesso la definizione "documentazione ufficiale". Questa è composta da quattro libroni formato A4 scritti dallo staff tecnico della Commodore / Amiga, comprendente, tra gli altri: Andy Finkel, Neil Katin, Dale Luck, Robert Mical, Jon Prince, Barry Walsh, Carolyne Scheppner, Robert Peck, Larry Hildenbrand, Bob Burns, Susan Deyl, Sam Dicker, Tom Pohorsky, Bruce Barret, Jim Mackraz, Bob Pariseau, Barry Whitebook e Stan Shepard, che, come si dice sul manuale "sono i busy guys che hanno creato Amiga e che sono Amiga".

I testi, tutti pubblicati dalla Addison Wesley, sono i seguenti:

Titolo: ROM Kernel Manual: Libraries and Devices
Pagine: oltre 800
Prezzo: $ 34.95

Titolo: INTUITION, Reference Manual
Pagine: circa 280
Prezzo: $ 24.95
Titolo: ROM Kernel Manual: Exec
Pagine: circa 200
Prezzo: $ 24.95

Titolo: Hardware Reference Manual
Pagine: circa 280
Prezzo: $ 24.95

Il primo descrive le routines grafiche e di supporto di Amiga ed i modi di interfacciamento col mondo esterno. Il secondo descrive l'interfaccia utente Intuition ad icone. Il terzo si occupa di Exec, cioè il sistema esecutivo multiprocesso di Amiga. Il quarto testo parla dei circuiti interni di Amiga e di come interagiscono con il sistema operativo.

Per procurarsi i volumi è necessario ordinarli presso una qualunque (buona) libreria

internazionale: i prezzi dovrebbero oscillare tra le 55.000 e le 90.000 lire.

Ogni libro è realizzato con grande cura e ricco di esempi: avvertiamo, però, che la loro piena comprensione richiede una quantità di tempo e di sforzi veramente sovrumana, come affermano anche colleghi americani, ovvero i redattori delle prestigiose riviste mondiali BYTE, PCW, Compute!. Questo per la quantità degli argomenti (192K di ROM più tutte le librerie esterne, le interfacce con i linguaggi e l'hardware), per la complessità dell'ambiente multitasking, incrementata dal processore 68000 (nuovo per la maggior parte dei dilettanti che arriva dal noto e confortevole mondo degli otto bit), dall'esistenza di tre circuiti VLSI originalissimi, dal fatto che il sistema è scritto in una mistura di Assembly 68000, BCPL e C quindi ci si trova ad armeggiare con sistemi di interfacciamento assolutamente nuovi, spesso discordanti e poco omogenei.

Il sistema operativo è ovviamente legato molto strettamente, ai livelli più bassi, allo hardware e quindi risente della sua complessità (25 canali DMA, canali asincroni, ambiente multiprogrammabile e multiprocesso, blitter che serve a spostare pixel ed a creare l'immagine video...), per questo anche il testo relativo allo hardware è d'obbligo.

I libri sono, a loro volta, strettamente collegati tra di loro. Ognuno, praticamente, comprende riferimenti ad argomenti discussi sugli altri tre. Diviene dunque difficile stabilire un ordine di importanza d'acquisto (o di lettura...) e, difatti, chi decide di occuparsi di Amiga acquista in blocco i quattro testi (con una spesa intorno alle duecentomila lire).

Non si può, praticamente, credere di sfruttare le funzioni e le caratteristiche dei livelli più "alti" del sistema (AmigaDOS ed Intuition), senza spesso trovarsi tra i piedi riferimenti alle routine macchinosissime dei livelli più bassi (Exec).

Inoltre, chi non conosce bene il linguaggio C, con cui sono spiegati i modi di interfacciamento e dialogo con le funzioni di sistema, si può tranquillamente scordare di programmare "a fondo" Amiga.

Titolo : L'AMIGA
Autore : Michael Bloom
Prezzo : lire 60.000

E' il primo testo per Amiga in lingua italiana degno di essere acquistato.

E' suddiviso in quattro sezioni per complessivi tredici capitoli. Si parla della "macchina", delle "immagini", dei "suoni" e delle "animazioni".

Il testo è rivolto ai neo-possessori ed ai probabili futuri possessori di Amiga, in quanto descrive molto spesso il "fenomeno" Amiga, dilugandosi nelle esaltazioni delle sue capacità grafiche e sonore.

La trattazione degli argomenti sembra piuttosto originale ,in quanto accanto alle descrizioni delle funzioni di Workbench e di AmigaBasic, si trovano ampie istruzioni per l'uso di programmi come Deluxe Paint, Deluxe Video, Deluxe Music Construction Set ed altri programmi commerciali a larga diffusione. La cosa, oltre a dare un'idea di che cosa sia effettivamente possibile fare con la macchina grazie al software già presente, può essere molto interessante per chi possiede magari i programmi descritti e non riesce a comprendere i ponderosi manuali originali in inglese (o non li ha...).

Si noti che la trattazione di AmigaBasic è tutt'altro che sistematica, ma rivolta più che altro a spiegare come si usano le speciali istruzioni per l'uso dei suoni, della grafica, delle finestre, di menu e sprites, presupponendo quindi la conoscenza del Basic (precisiamo che non è un manuale su AmigaBASIC).

Il libro è riccamente illustrato adatto a tutti meno che ai programmatori più esperti; ma per questi ci sono decine di altri testi.

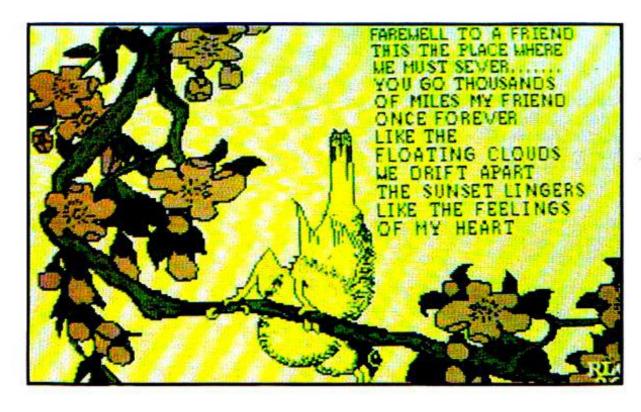
Titolo: AMIGA PROGRAMMER'S HANDBOOK (I)
Autore: Eugene P. Mortimore
Editore: Sybex
Codice: ISBN 0-89588-367-8
Prezzo: $ 24.95

Un testo di 635 pagine, rappresentante un dettagliato compendio del sistema operativo di Amiga. Il testo è aggiornato alla versione 1.2 del sistema operativo (dovrebbe esser disponibile anche l'edizione per la 1.3, ma non abbiamo notizie chiare in merito) e comprende la descrizione di alcune funzioni e caratteristiche non decritte nei testi originali della Commodore / Addison Wesley.

Sono elencate oltre 300 funzioni in ordine alfabetico, per classi e librerie: Graphics & Drawing, Animation, Text, Layer, Intuition, Workbench. Ogni capitolo descrive con grande precisione, anche se necessariamente con sinteticità, le caratteristiche del sistema, trattando quindi argomenti affascinanti, misteriosi e nuovi come: Bitmap, Icone, Multitasking, Layers, Gestione della memoria, finestre modi grafici speciali (Extra Half Brite, HAM, Dual Playfield) eccetera.

Per ogni funzione è spiegata, a grandi linee, il modo d'uso, i parametri necessari ed eventuali valori restituiti. Gli esempi sono praticamente inesistenti.

Il testo è chiaramente rivolto a chi ha una perfetta conoscenza del linguaggio C, una buona padronanza del sistema Amiga e, possedendo già la documentazione ufficiale della Commodore / Addison-Wesley, vuole avere un testo supplementare per scrivere programmi di altissimo livello. Pare che sarà presto tradotto in lingua italiana, ma data la complessità e la novità dell'argomento (zeppo di parole non traducibili e comunque complicate da spiegare) non vorremmo essere nei panni dei traduttori.

Siamo in attesa di leggere il secondo volume americano, dedicato alle librerie ed agli esempi d'uso, per poterlo descrivere.

UN'ENCICLOPEDIA PER IL TUO COMMODORE?

Straordinario!

Commodore Computer Club ti offre un'eccezionale combinazione, valida fino ad esaurimento delle scorte.

Ai nuovi lettori, infatti, offre gli 11 fascicoli dell'intera annata 1988, dal n.49 (gennaio) al n.59 (dicembre) inclusi, al favoloso prezzo di L. 49000, spese di spedizione comprese.

Per usufruire della fantastica offerta, invia subito la cifra richiesta mediante assegno bancario (non trasferibile), intestato a:

Systems Editoriale
Servizio arretrati
Viale Famagosta, 75
20142 MILANO

A causa dell'eccezionalità dell'offerta, gli abbonati non possono usufruire del consueto sconto a loro riservato.

Ricorda che i numeri arretrati di una pubblicazione tecnica, come la nostra, sono spesso più utili di una "vera" enciclopedia: articoli di informazione generale, risposte ai lettori, listati, tecniche di programmazione, didattica, utility, giochi... sono soltanto alcuni degli argomenti costantemente affrontati sulle pagine di "Commodore Computer Club" e risultano perfettamente comprensibili ai principianti ed utilissimi agli utenti evoluti.

N.B.: dal momento che i numeri arretrati sono in fase di rapido esaurimento, ricorda di indicare, nella lettera di accompagnamento, i numeri dei fascicoli alternativi che desideri, nel caso non sia possibile inviarti l'intera annata.

Titolo :PROGRAMMER'S GUIDE TO THE AMIGA
Autore : Robert A. Peck
Editore: Sybex
Codice : ISBN 0-89588-310-4
Prezzo : $ 24.95

Un libro d'obbligo per tutti i possessori di Amiga che vogliano cimentarsi nella programmazione in Pascal, Assembly e, particolarmente, in C sfruttando a fondo le caratteristiche del sistema operativo. Il libro sovrabbonda di esempi e di programmi, spesso semplici ma sempre esaurienti e chiari. Pregio maggiore di questo testo, crediamo, sia quello di essere stato scritto da qualcuno che abbia effettivamente lavorato con Amiga, quindi sa quali e quanti problemi incontra un programmatore, dilettante od evoluto, nell'avvicinarsi all'originale e nuovo mondo del kernel di Amiga. Peck, infatti, è uno degli ingegneri che hanno collaborato alla creazione di Amiga, ed autore del "ROM kernel Manual" e dello "Hardware manual", documentazione ufficiale Commodore.

Per la prima volta, dunque, in un testo si fornisce non solo la descrizione ed il funzionamento di alcune routines del sistema operativo, ma anche degli esempi di uso pratico, quasi sempre in Amiga (Lattice) C. I rimandi alla documentazione ufficiale non sono molto frequenti. Comunque, per una trattazione completa ed approfondita, tali testi sono sempre insostituibili.

L'opera di Peck, lunga 350 pagine, prevede nove capitoli: Overview del sistema, AmigaDOS, Exec, Graphics, Intuition, Devices, Animation, Sound, Multitasking. Le appendici parlano di ED, di Amiga C.

Crediamo che il testo sia veramente prezioso per tutti i programmatori evoluti, anche se siamo convinti che la documentazione ufficiale sia sempre una condizione vitale per studiare e programmare come si deve.

Titolo: THE AMIGADOS Manual
Autore: Commodore
Editore: Bantam Computer Books
Prezzo: $24.95

Si tratta della versione commerciale dei tre testi "ufficiali" supplementari rilasciati dalla Commodore su AmigaDOS insieme ai propri "Developers Pack", ovvero a quei package di programmi (compilatori Assembler, C, librerie ed utilities varie) e documentazione forniti alle software house che vogliono sviluppare programmi per Amiga in USA.

I tre testi inclusi sono chiamati: AmigaDOS User's Manual, AmigaDOS Technical Reference Manual, AmigaDOS Developer's Manual. Riferimenti a questi libri sono frequentemente rintracciabili sia nei quattro testi "fondamentali" della Addison Wesley, sia negli articoli su riviste estere, sia in tutti i libri su Amiga e perfino nei manuali acclusi alla macchina.

Sono qui contenute tutte le informazioni necessarie per usare i comandi di CLI, in modo assai sintetico ed asettico (anche peggio dei nostri articoli pubblicati su CCC in passato...). Inoltre sono descritti, per i programmatori più evoluti, le strutture fisiche dei files, settori e blocchi sul disco, però spesso si trovano agganci alla documentazione ufficiale. Inoltre è descritto l'uso del Macro Assembler 68000 della Commodore, del Linker ALINK, nonchè l'eventuale uso della coppia Amiga-Sun (una stazione di lavoro grafica).

Tale testo crediamo che sia veramente utile solo ai programmatori che abbiano già conoscenza di Amiga o che siano laureati in informatica od ingegneria; costoro, tuttavia, avranno difficoltà a capire le spiegazioni sintetiche delle strutture dei file, anche per la quantità di concetti assolutamente nuovi ed originali per i più (quanti hanno già studiato ambienti multiprogrammabili?). Gli altri possono accontentarsi delle rigorose descrizioni dei comandi del CLI e dell'assemblatore ma se non si conosce la struttura del sistema operativo, programmare in Assembler con Amiga è fantascienza; non per niente anche la documentazione usa sempre il C.

Tale testo è stato tradotto in italiano, con un lavoro lodevole, ma con qualità non sempre sufficiente specie nella traduzione di alcuni passaggi ingarbugliati, che richiederebbero la buona conoscenza, da parte del traduttore, dell'argomento trattato (cosa, in Italia, riservata a pochissimi). Il prezzo risulta di L. 54.000. Se qualcuno comincia a pensare che siamo contrari alle traduzioni di testi così tecnici ed impegnati, preferendo comunque scervellarci sugli originali, bè ha ragione! Si consideri che chi scrive il testo originale ha almeno le idee chiare e spesso l'argomento è complicato ed incomprensibile egualmente. Figuriamoci leggendo la traduzione fatta da chi non conosce l'argomento (perchè non può conoscerlo) ed "interpreta" senza senso.

Per chi non conosce nemmeno un po' la lingua inglese, tuttavia, è già meglio di niente.

QUATTRO CHIACCHIERE CON AMIGABASIC

Una scusa per parlare dell'Amigabasic e scoprirne le grandi potenzialità

di **Valentino Spataro**

Come per tutti i cambiamenti, il passaggio dal C/64 all'Amiga non è certo indolore. Ma che cosa ne giustifica il passaggio, e in che cosa consiste il dolore? Affrontiamo questi problemi senza discorsi teorici o complicati, ma partendo da un esempio concreto e dalle esperienze che se ne possono trarre.

LA TECNICA USATA

Il programma di queste pagine fa ripetere, al computer, una frase a piacere con la possibilità di variare i parametri della voce. Chi si aspettava una qualche eclatante novità sarà forse rimasto deluso: vi sono già molti programmi che permettono di far parlare l'Amiga. Tra gli altri, lo stesso programma SAY, fornito tra le utilities del disco Workbench, e Speech (tra i demos nel dischetto extras): per la versione 1.2 i due programmi sono presenti entrambi sul disco Extras.

Le differenze esistenti tra Say, Speech e

quello proposto sono diverse.

Anzitutto c'è da considerare lo scopo essenzialmente didattico di quest'ultimo: non si parlerà, infatti, solo del comando SAY (anzi, la sua trattazione sarà marginale; si rimanda all'articolo comparso su CCC n. 44 pagine 34 - 36 e al manuale) ma anche di come usare i famigerati (e professionali) menù pulldown o "a tendina". Parleremo, inoltre, di come aggiungere comandi al Basic standard, come usare il mouse con semplicità e, soprattutto, come implementare un potentissimo comando con cui tracciare, sullo schermo, barre di vario tipo grazie alle quali regolare (come si fa con gli equalizzatori degli impianti hi-fi) il livello di un "qualcosa" da definire a piacere.

Nel programma di queste pagine il metodo è usato per impostare livelli di velocità, di frequenza e di inflessione della frase da pronunciare.

Il programma, infine, è predisposto per creare simpatici effetti di eco, spesso usati nelle canzoni. Il listato, insomma, è tutto da leggere (e da usare).

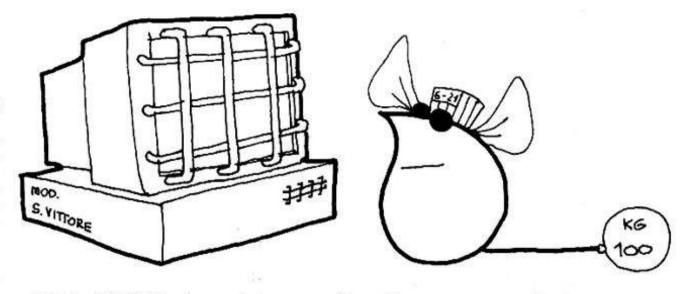
PROGRAMMI E STANDARD ASCII

Forse non tutti sanno che il programma può essere scritto sia in ambiente Abasic sia mediante Notepad o qualsiasi altro wordprocessor purchè registri i file in formato Ascii. Solo per far girare il programma è necessario caricarlo da Amigabasic.

Bisogna ricordare, però, che Notepad non genera un'icona tale da permettere di richiamare automaticamente l'Amigabasic cliccando con il mouse; in questo caso, infatti, viene richiamato Notepad, vale a dire il programma che ha generato il file Ascii.

Dopo aver, però, caricato il file da Amigabasic (e registrato nuovamente), verrà generata l'icona Abasic - compatibile. In proposito è bene sottolineare che un programma Basic può essere memorizzato su dischetto in tre modi diversi. Il primo si realizza con il comando...

SAVE "nomefile", x

...in cui, con x=A, salveremo il file in formato Ascii; con x=B in formato binario (leggibile solo da Amigabasic); con x=P in un formato non più listabile. Riteniamo utile sottolineare la comodità offerta dallo standard Ascii. I listati, infatti, possono esser scritti utilizzando programmi anche profondamente diversi tra loro, come wordprocessor (Scribble, Notepad), compilatori, oppure, addirittura, provenienti (via cavo seriale, modem o dischetti) da personal

computer MS-DOS (e viceversa).

Esistono, infatti, in commercio alcuni programmi che permettono di trasferire i file Ascii registrati in formato Amiga su dischetti 3.5 formattati in formato MS-DOS. Caricando i file così trattati su un PC compatibile, che disponga di un drive da 3.5 pollici, è possibile far girare gli stessi programmi anche su macchine Ms-Dos, sempre che non siano state usate istruzioni presenti solo nell'AmigaBasic o con sintassi incompatibile col il Basic fornito con i PC.

IL PROGRAMMA

Il programma è di uso intuitivo. Fatto partire, richiede la digitazione della frase che dovrà esser poi pronunciata. Subito dopo viene caricato, da Workbench, un programma chiamato "Narrator Device" necessario per far parlare Amiga; inserite, quindi, il disco Workbench quando richiesto. Pronunciata la frase, comparirà un grafico con 10 colonne su ognuna delle quali è indicato il valore impostato di default. Le prime nove indicano i corrispondenti parametri descritti esaurientemente a partire dalla pagina 211 del manuale AmigaBasic: l'ultima verrà spiegata più avanti.

Per prendere maggiore confidenza con il programma, vi consigliamo (dopo aver digitato la frase e premuto Return) di:

• posizionare il puntatore del mouse all'interno del rettangolo contenente le barre e premere il tasto sinistro del mouse;
• tenendolo premuto spostatevi all'interno della finestra: potrete così variare gli indici, e relativi parametri, con stupefacente semplicità. Ad OGNI variazione verrà ripetuta automaticamente la frase.

Passiamo ora alla realizzazione degli effetti speciali. Proviamo, ad esempio, a riprodurre dapprima una frase al volume massimo e poi, pian pianino, sempre più basso.

Basterà posizionarsi con il puntatore sulla parte alta della colonna del volume (la sesta) e premere il tasto sinistro del mouse. Tenendolo premuto, e scendendo lentamente verso il basso (più lunga è la frase, più lento deve essere il movimento) otterrete il simpatico effetto. Impostando, poi, la nona colonna alternativamente a 0 e a 2 il programma aspetterà che la frase sia pronunciata per intero prima di proseguire (se impostato a 0) oppure procederà nella elaborazione (se impostato a 2) interrompendo la pronuncia.

L'ultima colonna è utilizzata per regolare il ritardo tra una frase e l'altra, regolando anche la velocità di "sfumatura" della frase stessa. Provate a ripetere analoghi esperimenti con le altre barre per scoprire effetti combinati. Inoltre è possibile registare qualsiasi frase (o suono) emessa da Amiga con comuni registratori, collegando questi ultimi direttamente all'uscita audio di Amiga.

UNO SGUARDO AL PROGRAMMA

Procediamo con ordine: innanzitutto consigliamo di osservare il listato per quanto riguarda l'uso dei menù pulldown e l'uso del mouse, la creazione di comandi nuovi e, in particolare, il comando BARRE.

Quanto ai menù pulldown, per le notizie tecniche rinviamo al manuale fornito con la macchina; cercando nel programma le istruzioni MENU vedrete come è semplice usarle. E non facciamo come al solito quelli che dicono: "Il C/64 non le aveva, quindi ne potrò fare anche a meno" per pura e semplice pigrizia! Osservando le prime tre righe del listato vedrete come è semplice crearli.

Ma veniamo a quello che certamente è il pezzo forte: il nuovo comando Barre. Innanzitutto consigliamo di (ri)leggere l'articolo a pagina 67 (CCC n. 58) sui comandi SUB o il manuale Basic in proposito.

Il comando Barre permette di scegliere un valore, tra un minimo e un massimo definiti, spostando un indice all'interno di una "barra". Lo spostamento avviene premendo il tasto sinistro del mouse una volta posizionato il puntatore all'interno della stessa barra.

La sintassi del comando è...

barre x, y, lx, ly, min, max, vl

..in cui:

- x, y indicano le coordinate dell'angolo superiore sinistro della barra;
- lx, ly le dimensioni della barra stessa;
- min, max l'intervallo dei valori tra cui scegliere;
- vl il valore corrente.

Se, per esempio, in un programma vogliamo emettere un suono di frequenza X compresa tra 50 e 15000 (da definirsi da parte dell'utente) basterà aggiungere al nostro listato le linee del programma proposto comprese tra "SUB barre ... STATIC" e "END SUB" eliminando la linea che inizia con SHARED e modificando la linea che inizia con "GOSUB disegna:" nel seguente modo:

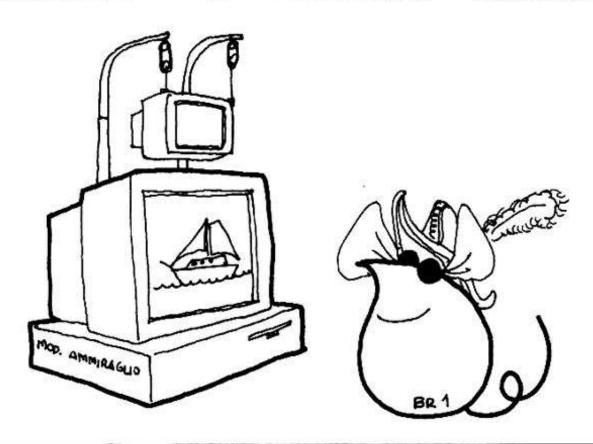
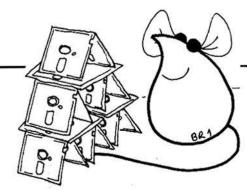
GOSUB disegna: sound vl, 10, 64: GOTO legge

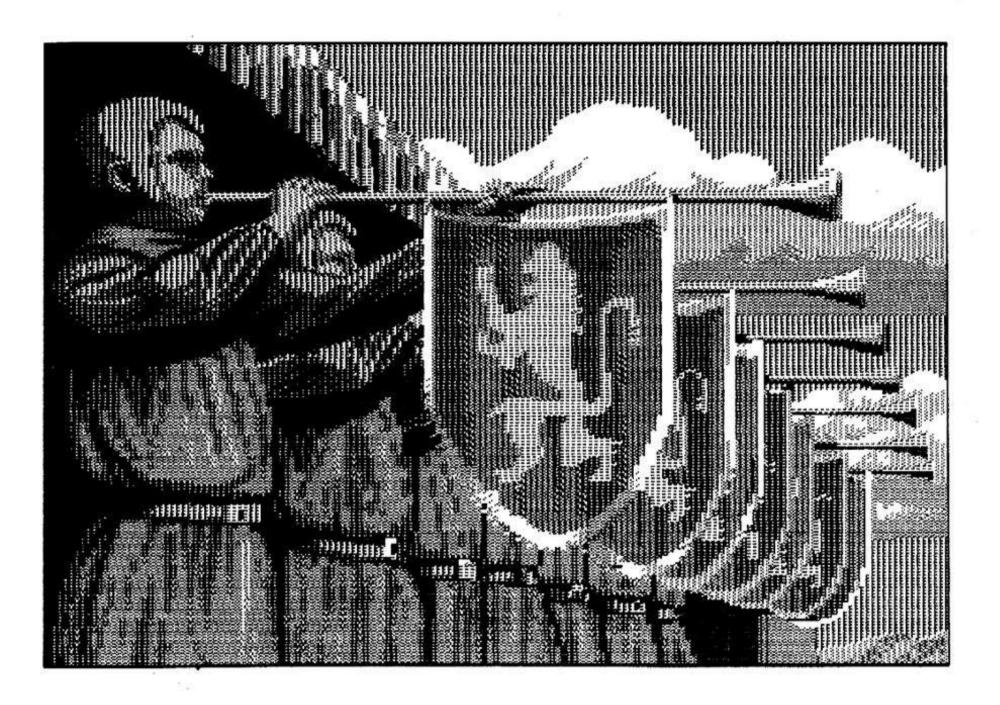
Quando necessario, scriverete, nel vostro programma...

barre 50!,50!,10!,100!,50!,15000!,7000!

Premendo il pulsante sinistro, al di fuori della barra, il controllo tornerà al programma principale.

Il comando permette di sapere immediatamente qual è il valore impostato o modificato e (tramite la modifica alla linea iniziante con "GOSUB disegna") di osservare immediatamente l'effetto finale.

Il programmino dimostrativo, presente in queste stesse pagine, consente al lettore principiante di esaminare in dettaglio la tecnica descritta.

```
-inizio:' demo per comando BARRE
  CLS:PRINT "demo per comando BARRE by V.Spataro"
  x=50:y=50:lx=20:ly=100:min=50:max=15000:vl=9000
  barre x,y,lx,ly,min,max,vl:GOTO inizio

 SUB barre (x,y,lx,ly,min,max,vl) STATIC
   y1=-((vl-min)*ly/(max-min))+ly+y:LINE (x,y)-(x+lx,y+ly),2,bf
   GOSUB disegna
 legge:
   IF f$="i" GOTO fine
   IF MOUSE (0)>-1 GOTO legge
 procede:
   x1=MOUSE (1):y1=MOUSE(2)
   IF x1<x OR x1>x+lx OR y1>y+ly OR y1<y THEN GOTO fine
   vl=y1:LINE (x,vy)-(x+lx,vy),2
 GOSUB disegna:SOUND vl,7:GOTO legge
 disegna:
   LINE (x,y1)-(x+lx,y1),1:vy=y1:vl=(max-min)*(ly+y-y1)/ly+min
   LOCATE y/8-1,x/8:PRINT vl;" ":RETURN
 fine:
   y1=-((vl-min)*ly/(max-min))+ly+y:LINE (x,y1)-(x+lx,y1),1:
   END SUB
```

.AMIGA PARLA

```
MENU 1,0,1,"project":MENU 1,1,1,"input parola"
MENU 1,2,1,"quit   ":MENU 2,0,0,"":MENU 3,0,0,""
MENU 4,0,0,"":MENU ON:ON MENU GOSUB topo
 lx=20:y=40:scala=60:ni=9:FOR a=0 TO ni:READ fr%(a):NEXT
 FOR a=0 TO ni:READ ly(a),min(a),max(a):NEXT
start:
  MENU ON:CLS:INPUT "frase ";fr1$:fr$=TRANSLATE$(fr1$):SAY fr$,fr%
  CLS:PRINT fr1$:LINE (0,y-17)-(ni*scala+lx*2+10,y+110),1,b
  FOR a=0 TO ni
   x=a*scala+lx:LINE (x,y)-(x+lx,ly(a)+y),2,bf
   y1=ly(a)+y-((fr%(a)-min(a))*ly(a)/(max(a)-min(a)))
   LINE (x,y1)-(x+lx,y1),1:LOCATE y/8-1,x/8:PRINT fr%(a);
  NEXT
scelta:
  IF f$="i" THEN f$="" :GOTO start
  IF MOUSE(0)>-1 GOTO scelta
   nt=CINT((MOUSE (1)+lx)/scala)-1:y1=MOUSE(2)
   IF y1<y-17 OR (nt<0 OR nt>9) THEN SAY fr$,fr%:GOTO scelta
   x=nt*scala+lx:vl=fr%(nt):max=max(nt):min=min(nt):ly=ly(nt)
   barre x,y,lx,ly,min,max,vl
 GOTO scelta

SUB barre (x,y,lx,ly,min,max,vl) STATIC
  y1=-((vl-min)*ly/(max-min))+ly+y:LINE (x,y)-(x+lx,y+ly),2,bf
  GOSUB disegna:SHARED f$,fr$,fr%(),nt:GOTO procede
legge:
  IF f$="i" GOTO fine
  IF MOUSE (0)>-1 GOTO legge
procede:
  x1=MOUSE (1):y1=MOUSE(2)
  IF x1<x OR x1>x+lx OR y1>y+ly OR y1<y THEN GOTO fine
  vl=y1:LINE (x,vy)-(x+lx,vy),2
GOSUB disegna:fr%(nt)=vl:SAY fr$,fr%:FOR a=1 TO fr%(9):NEXT:GOTO legge
disegna:
  LINE (x,y1)-(x+lx,y1),1:vy=y1:vl=(max-min)*(ly+y-y1)/ly+min
  LOCATE y/8-1,x/8:PRINT vl;" ":RETURN
fine:
  y1=-((vl-min)*ly/(max-min))+ly+y:LINE (x,y1)-(x+lx,y1),1:
  END SUB
'valori di default per comando say (v. manuale basic pg.211)
DATA 110,0,150,0,22200,64,10,1,0,10
' valori di ly,min,max per grafico a barra
DATA 100,65,320,10,0,1,100,40,400,10,0,1,100,5000
DATA 28000,64,0,64,11,0,11,10,0,1,10,0,2,100,0,1000
topo:i1=MENU(0):i2=MENU(1)
     IF i1=0 AND i2=2 THEN MENU RESET:END
     IF i1=0 AND i2=1 THEN f$="i":PRINT "ok":RETURN
GOTO topo
```

UNA STAMPANTE, MOLTI COMPUTER

La nuova Commodore MPS 1230, piccola e compatta, evidenzia notevoli potenzialità, tra cui la possibilità di collegarla subito con qualsiasi computer

di **Stefano Simonelli**

La nuova stampante Commodore MPS 1230 consente di utilizzare la modalità di scrittura DRAFT, a definizione normale, oppure NEAR LETTER QUALITY, ad alta definizione. Entrambe possono essere realizzate con diverse spaziature orizzontali e in numerose versioni, riportate nelle tabelle di queste pagine.

La MPS 1230 possiede così tante combinazioni di resa grafica che uno stesso carattere può essere stampato in ben 41 modi diversi.

Una sua caratteristica interessante è quella di programmare, anche senza usare computer, tutte le funzioni di cui dispone, davvero numerose; infatti, accendendola mentre si tengono premuti conteporaneamente i tasti LINE FEED e FORM FEED, la stampante si predispone per essere programmata (modo SET UP).

A questo punto vengono proposte, una per una, tutte le predisposizioni possibili; ad ognuna l'operatore dovrà rispondere premendo il tasto LINE FEED (equivalente a SI') per accettare la predisposizione oppure FORM FEED (=NO) per rifiutarla.

La programmazione può venire interrotta in qualsiasi momento premendo il tasto LOCAL; è inoltre possibile annullarla del tutto, o confermarla, rifiutare o accettare l'ultima proposta.

La stampante può essere, ovviamente, programmata anche via software, utilizzando precise sequenze di escape descritte dettagliatamente nel manuale; i parametri inviati dal computer verranno persi solo nel momento in cui verrà tolta l'alimentazione alla MPS 1230; al contrario, settandola dal modo SET UP, le predisposizioni non verranno "dimenticate" neppure dopo aver tolto l'alimentazione, ma solo dopo averla riprogrammata secondo le stesse modalità.

I TEST DI STAMPA

Vi sono altre due funzioni settabili al momento dell'accensione: la prima è accessibile tenendo premuto il tasto LINE FEED (entrando così in modo TEST DI STAMPA nel quale sarà "plottato" l'intero set di caratteri). La seconda, attivabile premendo FORM FEED, imposta la macchina in modalità DATA SCOPE, procedura che la "forza" a stampare, in esadecimale, i dati inviati dal computer.

L'MPS 1230 può operare anche in modo grafico secondo il metodo Bit Image Mode (BIM).

Un carattere Bit Image viene definito mediante una configurazione di punti all'interno di una griglia di 7 righe per 16 colonne, che corrisponde alla testina di stampa. Questa stamperà l'immagine grafica desiderata azionando gli aghi corrispondenti ai punti definiti nella griglia stessa; sarà così possibile ottenere figure, grafici, diagrammi ed ogni tipo di disegno riproducibile punto-punto.

La stampa può essere effettuata su fogli di diversi tipi, nelle più svariate dimensioni

```
10 SCREEN 3
20 CLS
30 FORI= 10 TO 200 STEP 10
40 LINE (0,I)-(I,200)
50 LINE (320,200-I)-(320-I,0)
60 NEXTI
80 LINE (100,100)-(220,100)
90 FORI= 40 TO 100 STEP 5
100 LINE (160,I)-(120+I,100)
110 NEXTI
120 FORI= 40 TO 100 STEP 5
130 LINE (160,I)-(200-I,100)
140 NEXTI
150 FORI= 100 TO 160 STEP 5
160 LINE (I,100)-(160,I)
170 NEXTI
180 FORI= 100 TO 160 STEP 5
190 LINE (320-I,100)-(160,I)
200 NEXTI
210 LINE (0,0)-(319,0)
220 LINE (0,0)-(0,199)
230 LINE (319,0)-(319,199)
240 LINE (0,199)-(319,199)
250 LCOPY1
```

```
10 SCREEN 3
20 CLS
30 FORI= 10 TO 200 STEP 10
40 LINE (0,I)-(I,200)
50 LINE (320,200-I)-(320-I,0)
60 NEXTI
80 LINE (100,100)-(220,100)
90 FORI= 40 TO 100 STEP 5
100 LINE (160,I)-(120+I,100)
110 NEXTI
120 FORI= 40 TO 100 STEP 5
130 LINE (160,I)-(200-I,100)
140 NEXTI
150 FORI= 100 TO 160 STEP 5
160 LINE (I,100)-(160,I)
170 NEXTI
180 FORI= 100 TO 160 STEP 5
190 LINE (320-I,100)-(160,I)
200 NEXTI
210 LINE (0,0)-(319,0)
220 LINE (0,0)-(0,199)
230 LINE (319,0)-(319,199)
240 LINE (0,199)-(319,199)
250 LCOPY1
```

```
10 SCREEN 3
20 CLS
30 FOR I=10 TO 100 STEP 10
40 CIRCLE (160,100),I
50 NEXTI
60 FOR I=160 TO 260 STEP 35
70 PAINT (I,100)
80 NEXT I
90 LINE (0,0)-(319,0)
100 LINE (0,0)-(0,199)
110 LINE (319,0)-(319,199)
120 LINE (0,199)-(319,199)
130 LCOPY1
```

```
10 SCREEN 3
20 CLS
30 FOR I=10 TO 100 STEP 10
40 CIRCLE (160,100),I
50 NEXTI
60 FOR I=160 TO 260 STEP 35
70 PAINT (I,100)
80 NEXT I
90 LINE (0,0)-(319,0)
100 LINE (0,0)-(0,199)
110 LINE (319,0)-(319,199)
120 LINE (0,199)-(319,199)
130 LCOPY1
```

(il formato massimo è il consueto A-4). Nella confezione sono presenti un introduttore di carta per fogli singoli ed uno sprocket (trattore dentato) utile per i moduli continui.

Decisamente agevole è il modo con cui si può inserire, o estrarre, il nastro inchiostrato, che evita il pericolo di sporcarsi le mani e consente una rapida comprensione del meccanismo di bloccaggio.

Il buffer di stampa consiste in una Ram, interna, di ben 4K che vengono utilizzati per immagazzinare dati durante la trasmissione.

Dopo un breve periodo iniziale, in cui 4K di dati vengono trasmessi senza interruzioni, si può continuare il lavoro mentre la macchina prosegue la stampa "svuotando" il buffer immagazzinato.

Altra nota di merito è sicuramente rappresentata dal manuale di istruzioni in dotazione, scritto in italiano, che risulta molto chiaro e di facile comprensione. Anche nei punti più critici, come il capitolo dedicato alla programmazione, è difficile incappare nelle tipiche difficoltà che, al contrario, si riscontrano con manuali di altre stampanti.

L'unico problema rilevato riguarda l'utilizzo dello sprocket: con fogli a modulo continuo molto lunghi, infatti, la carta, a volte, si accartoccia, ma solo nel caso in cui il pacco di carta non risulta perfettamente allineato con i rulli dentati; sistemando opportunamente il modulo continuo, invece, l'inconveniente non si verifica.

E' doveroso ricordare che, tuttavia, anche senza l'estrattore si riesce a stampare perfettamente, grazie soprattutto alle guancette guidacarta (di cui la stampante è fornita) che riescono a tenere il foglio perfettamente allineato.

CONCLUSIONI

L'MPS 1230 non è una stampante dedicata esclusivamente al C/64-128; è infatti munita sia di porta seriale (standard C/64, C/128, C/16, Plus/4, Vic/20) sia di porta parallela (standard Centronics) tramite la quale collegarla all' AMIGA o ad un qualunque compatibile IBM.

Le prove effettuate l'hanno vista impegnata con un C/128-64, un AMIGA ed un T-1200 Toshiba: il risultato è stato decisamente esaltante poichè, nei tre casi, ha svolto il lavoro in modo impeccabile sia con svariati pacchetti di word-processor sia con programmi di grafica.

La compatibilità con lo standard 803, che più sta a cuore ai lettori utenti dei "piccoli" Commodore, è totale, come è possibile rilevare dagli esempi pubblicati.

Anche le varie cartucce che consentono hard copy di schermate in alta risoluzione non hanno evidenziato difficoltà di alcun tipo.

TO CONFIRM PRESS LF, TO CHANGE PRESS FF, TO END PRESS LOCAL

INTERFACE
SERIAL COMMODORE

PRINTER EMULATED IN PARALLEL AND SERIAL COMMODORE
EPSON FX 80

CHARACTER SET IN PARALLEL MODE
ITALY SPAIN U.S.A. FRANCE GERMANY U.K. DENMARK SWEDEN ITALY

CHARACTER SET IN COMMODORE MODE
ITALY

OPEN MODE
5 COMMODORE COMMANDS 4 COMMODORE COMMANDS

AUTOMATIC SHEET FEEDER
NO YES

DOUBLE STRIKE PRINTING
BIDIRECTIONAL

WOULD YOU LIKE TO STORE THESE PARAMETERS ?
YES

CHANGES EXECUTED

Pica	10	car/pollice
Elite	12	car/pollice
Micro	15	car/pollice
Condensed	17.1	car/pollice
Pica condensed	20	car/pollice
Elite condensed	24	car/pollice
Micro condensed	30	car/pollice

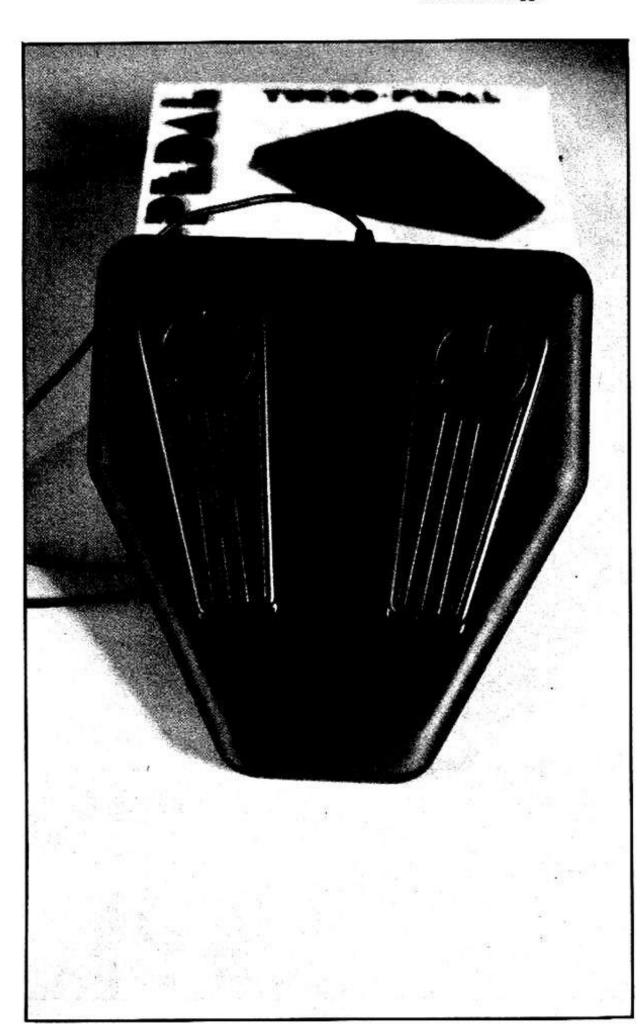
Spaziature orizzontali della Mps 1230

Grassetto	(emphasized)
Doppia larghezza	(double-wide)
Apici	(superscript)
Pedici	(subscript)
Doppia passata	(double strike)
Sottolineatura	(underline)

Tipi di caratteri disponibili
sulla Commodore Mps 1230
ed attivabili con semplici ESC

UN COMPUTER A PEDALI

Per alcuni giochi di simulazione il solito joy non basta

di **Michele Maggi**

Tutti noi abbiamo giocato, almeno una volta, con i portentosi videogames che ci fanno illudere di pilotare una rombante motocicletta o una velocissima Ferrari.

Di solito il movimento del veicolo viene affidato, nelle sempre più realistiche simulazioni, agli accessori più diffusi sul mercato; ci riferiamo, ovviamente, ai joystick (soprattutto nel caso del C/64) ed al mouse (quasi esclusivamente per Amiga).

Alcune manovre dei mezzi, però, devono essere sviluppate mediante il movimento avanti - indietro dell'assicella che, nella realtà, è del tutto diverso dalla pressione dei pedali che è necessario effettuare nei veicoli "veri".

Per cambiare direzione lo spostamento a destra ed a sinistra può essere accettabile, ma l'accelerazione e la frenata trovano poca rispondenza con gli altri due movimenti del joy.

A questo inconveniente hanno ovviato i progettisti di *Turbo - Pedal*. Questo accessorio, particolarmente robusto e dotato di ventose che lo ancorano al pavimento, dispone di due pedali che, in effetti, altro non sono se non due *switch* collegati con quelli corrispondenti, in un comunissimo joystick, alle posizioni avanti e indietro.

Sul retro di *Turbo Pedal* troviamo il consueto cavetto per il collegamento con il computer (rigorosamente C/64 e Amiga compatibile) ed un secondo connettore nel quale va inserito un comunissimo joystick.

In pratica, il joy si inserisce nella "pedaliera" e lo spinotto di quest'ultima nel connettore del computer.

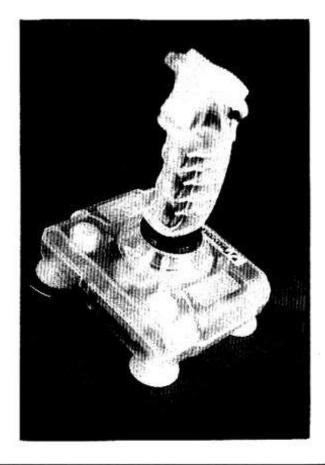
All'interno di Turbo Pedal un particolare collegamento impedisce, al computer, di riconoscere il movimento avanti - indietro del joy, ma gli consente di individuare solo la pressione dei pedali, permettendo quindi al giocatore di concentrarsi al massimo nel videogame.

Il risultato? Per quanto riguarda i giochi di simulazione di movimento (tra cui, soprattutto, il pilotaggio di macchine, moto ed aerei) i vantaggi sono enormi: con il nuovo accessorio è possibile raggiungere punteggi impensati grazie alla naturalezza con cui il giocatore è finalmente in grado di governare i potenti bolidi.

UN DOPPIO AUTOFIRE

Un joystick di nuova concezione è stato di recente presentato sul mercato. Si tratta di un joy dotato di vari tasti *fire* (per consentire al giocatore una scelta più ergonomica) ma, soprattutto, permette di simulare, elettronicamente, altri due effetti, non comuni in accessori del genere.

Il primo è il famoso *Autofire*, vale a dire un circuito in grado di simulare, in rapida successione, la pressione del tasto Fire senza esser costretti a slogarsi l'indice (e a rovinare il joy) a causa di ripetute pressioni sul corrispondente pulsante: è sufficiente, ad esempio, tenerlo premuto per veder partire, l'uno dopo l'altro, decine di proiettili dalla pistola laser dei vostri eroi preferiti.

L'altro effetto è del tutto simile all'autofire, ma chiama in causa la simulazione del continuo spostamento destra - sinistra dell'assicella del joystick. Basta quindi tener premuto il pulsante per raggiungere punteggi incredibili in giochi tipo *Decathlon* in cui, con un joy normale, ci vedremmo impegnati a scomodi e faticosi spostamenti dell'assicella.

Inutile dire che è presente un deviatore a tre vie: nella prima posizione qualsiasi automatismo viene escluso, ed il joy si comporta nel modo standard. Nella seconda posizione viene attivato l'autofire ed ogni volta che si preme il pulsante superiore verrà simulata la pressione discontinua del pulsante fire. Nella terza posizione, infine, la pressione del pulsante simulerà il movimento sinistra - destra del joy.

L'accessorio è disponibile in vari colori, tra cui un'originale versione trasparente.

Per informazioni:
Digimail S.r.l.
Via Coronelli, 12
20146 Milano - Tel. (02) 426559

QUANTO COSTA UNA VACANZA

Siamo spesso abituati a determinare una somma, conoscendo gli addendi; può capitare, invece, di dover individuare un addendo conoscendo la somma

di **Alessandro de Simone**

Per alcuni le vacanze sono già finite; per altri, invece, devono ancora iniziare. In entrambi i casi, però, sicuramente è stata sperimentata la fase (spesso critica) della cifra da spendere per trascorrere il periodo di ferie.

Chiunque di noi ha effettuato una scelta basata su solide argomentazioni:

"Quell'albergo chiede 70 mila lire al giorno, ma non comprende, tra i servizi, quello di spiaggia. Quell'altro chiede 63 mila, compresi ombrellone e sdraio, ma la sua distanza dalla spiaggia costringe ad usare la macchina. Qull'altro ancora chiede 80 mila, ma offre lo sconto del 30% ai bambini..." e così via.

Per risolvere il problema, bisognerebbe impostare correttamente uno *Spread sheet* (foglio elettronico di ricalcolo automatico) in cui ogni colonna rappresenta un albergo e ogni riga una delle "voci" di spesa (vitto, alloggio, servizi di spiaggia, mance, viaggio, necessità di eventuale traghetto, numero di giorni eccetera). In questo modo, però, è possibile "solo" conoscere la somma giornaliera da sborsare in corrispondenza di ciascun albergo. Non è possibile (almeno con gli spreadsheet più comuni) effettuare automaticamente il calcolo inverso.

LE FORMULE INVERSE

Tutti noi, a scuola, ci siamo imbattuti almeno una volta in una formula inversa. Supponendo che...

Somma = A + B + C

...come fare a sapere il valore "B" se conosciamo i dati rimanenti?

La risposta, che riportiamo solo per completezza (tutti noi sappiamo di che si tratta, vi pare?), è:

B = Somma - A - C

Il ricorso alle cosiddette *formule inverse* consente quindi di rispondere a domande del tipo: "Volendo spendere in media, per ogni giorno di vacanza, 80 mila lire, comprendendo, in tale media, 10 mila di benzina e di autostrada, e 7 mila di spese varie, quanto deve costare l'albergo?".

Oppure: "Volendo spendere 75 mila di albergo e 12 mila di spese varie, quanto posso disporre per benzina (ed autostrada)? In pratica, di quanto mi posso allontanare da casa senza eccedere nelle spese?".

La metodologia da seguire, nel semplice caso prospettato, deve tener conto del fatto che un addendo, spostandosi da destra a sinistra del segno *eguale* (=), cambia di segno; tutti gli altri, ovviamente, rimangono identici. Gli "spostamenti" degli addendi, quindi, devono essere individuati con la massima attenzione perchè, come è noto, basta un segno positivo o negativo errato per fornire risultati inattendibili.

IL PROGRAMMA

Il primo, brevissimo programma (*Sommatoria e formule inverse*) consente di risolvere la sommatoria qualunque sia l'incognita. Dopo il *Run* compare la formula che il listato è in grado di elaborare...

A + B + C + D + E + F = G

Subito dopo compaiono 7 domande relative al valore da attribuire a ciascuna variabile; l'incognita deve essere indicata semplicemente digitando "X" in corrispondenza della relativa domanda.

Provate a rispondere nel modo seguente...

A = ? 10
B = ? 20
C = ? 30
D = ? 40
E = ? 50
F = ? 60
G = ? X

VORREI, MA (NON) POSSO

La ricerca automatica delle formule inverse, una volta impostata quella di "base", è uno degli algoritmi più impegnativi per un programmatore evoluto.

E' infatti richiesto un lavoro non indifferente per costringere il calcolatore ad "estrarre", con la disinvoltura propria di noi... umani, una certa incognita da un contesto matematico più generale.

Si tratta, inevitabilmente, di "interpretare" la formula stessa, effettuare i dovuti trasferimenti a sinistra o a destra del segno eguale, tenendo conto delle rigide regole che tali trasferimenti comportano. Infine, dalla formula inversa così ottenuta, interpretare nuovamente la simbologia (livelli di parentesi, divisioni, raccolta a fattor comune) ed individuare, se necessario, i casi particolari che indurrebbero il computer in errore: divisione per zero, estrazioni di radici negative, logaritmi di valori negativi e così via, in una casistica difficile da stabilire a priori.

Il problema è decisamente arduo e, chi arriva alla fine, può ritenersi un vero e proprio genio della programmazione.

Pur non essendo geni, ma limitandosi ad essere... eccezionali, è possibile delimitare la vastità tipica di algoritmi tanto complessi.

Ci si potrebbe limitare, ad esempio, a determinare le formule inverse di relazioni matematiche che non contengono più di quattro variabili, oppure che non contengono parentesi, oppure che non prendano in considerazione altre funzioni, se non le quattro operazioni.

Spesso ci sentiamo dire, da lettori che ritengono di esser molto preparati, che gli argomenti affrontati da *Commodore Computer Club* sono troppo banali per il loro livello di esperienza.

Bene, ora tocca a noi lanciare la sfida: C'è qualche aspirante genio che sia in grado di scrivere un programma che, a partire da una *qualsiasi* relazione matematica, "estragga" *tutte* le formule inverse?

Noi dubitiamo che esistano persone "normali" capaci di tanto: ma siamo, ovviamente, altrettanto desiderosi di esser smentiti clamorosamente.

Volete partecipare alla sfida proposta?

Mi pare già di vedere qualcuno, tra voi, che annuisce beffardo...

...premendo il tasto Return dopo ogni valore. La risposta, scontata, è...

G = (X) = 240

La somma totale, infatti, vale 240. Subito dopo il cursore lampeggia nuovamente sulla prima domanda per consentire un'altra elaborazione. Provate, allora, a rispondere nel modo seguente...

A = ? 10
B = ? X
C = ? X
D = ? 40
E = ? 50
F = ? 60
G = ? 300

...limitandosi a premere il tasto Return nei casi in cui la cifra che compare (A, D, E, F) non richieda modifiche.

In tal modo, in pratica, avete chiesto al programma di rispondere alla seguente domanda: Quale deve essere il valore della somma di "B" e di "C" affinchè (tenendo conto dei rimanenti valori) la somma totale sia 300?

La risposta non si fa attendere: 140. Se, alla successiva digitazione, lasciate inalterati tutti i valori (ad eccezione di "C", cui assegnate il valore 100) la risposta diventerà 40, come ovvio.

Il programma, quindi, è in grado non solo di individuare una qualsiasi incognita, ma anche il valore globale che deve possedere un "gruppo" di incognite.

E' prevista una segnalazione di errore nel caso in cui (righe 210 - 240) nessuna "X" compaia nella digitazione. La stringa X$, contenente solo spazi (vedi riga 140), è incaricata di cancellare sia il messaggio di errore sia l'elaborazione precedente (riga 280), nel caso questa contenga più cifre di quella più "recente".

Le tre righe, da 250 a 270, contengono la vera e propria elaborazione da effettuare; tutte le altre, infatti, sono destinate all'input, al corretto incolonnamento e all'individuazione di eventuali errori.

NON SOLO SOMME

Il breve programma precedente si limita a determinare una qualsiasi incognita della sommatoria di sei addendi. Nel caso in cui si volesse un maggior numero di addendi, è sufficiente manipolare gli indici del For ... Next (righe 170, 190, 210, 220, 250, 270) prestando attenzione a non esagerare per evitare fenomeni indesiderati di *scrolling*.

Che fare, però, se la formula "base" da elaborare (da cui, cioè, ricavare le formule inverse) non è una semplice sommatoria, oppure contiene una divisione, una radice quadrata o una funzione logaritmica?

La risposta, purtroppo banale, implica la scrittura di un programma *ad hoc*, caso per caso.

Ne è la prova il secondo listato pubblicato in queste pagine ("Percentuali e formule inverse") che consente di determinare uno dei tre parametri (QI, quantità iniziale; QF, quantità finale; P, percentuale) essendo noti gli altri due.

La formula base considerata...

Qf = Qi (1 + P / 100)

...è incaricata di risolvere i tre problemi proposti. Un tipico esempio è il seguente: "Quale somma ritirerò dalla banca se vi deposito 1 milione al tasso del 5%?"

X = 1000000 x (1 + 5 / 100)
X = 1005000

Inutile dire che la procedura da seguire è la stessa del programma precedente. E' sufficiente digitare "X" in corrispondenza dell'incognita cercata, ed i valori noti ove necessario.

Da segnalare la presenza di due messaggi di errore. Il primo (riga 250) compare nel caso in cui non viene indicata l'incognita; il secondo (righe 320) impedisce l'elaborazione nel caso in cui, alla variabile "QI", sia attribuito il valore nullo. Si evita, in altre parole, la divisione per zero (riga 330).

Da quest'ultimo particolare risultano ancora più evidenti le difficoltà che si incontrano nello stabilire un corretto algoritmo di elaborazione dovendo prevedere tutti i casi possibili.

Il programma, tuttavia, risulta sufficientemente "aperto" e suscettibile di consistenti modifiche da parte dell'utente.

PRINCIPIANTI, I DIECI COMANDAMENTI

1 Leggi attentamente il libretto di istruzioni del tuo computer e delle periferiche che ti sei procurato (registratore, drive, stampante, monitor, eccetera)

2 Non limitarti a leggere i listati di esempio ivi riportati, anche se sembrano banali: digitali sul computer e falli girare (digitando Run e premendo il tasto Return). Se non ti sono chiari, al contrario, la loro digitazione (e successiva esecuzione) ti chiarirà le idee.

3 Nel digitare i listati, ricordati di premere SEMPRE il tasto Return quando giungi alla fine del rigo, anche se tale operazione può sembrarti inutile.

4 Non confondere la vocale alfabetica "O" con il numero zero "0".

5 Digita sempre per esteso il comando PRINT e non abbreviarlo mai con il punto di domanda (?).

6 Inizia a digitare i listati più semplici e brevi: quelli più lunghi potrai digitarli quando avrai acquisito una maggior dimestichezza con il computer.

7 Dopo aver digitato un QUALSIASI programma, registralo subito, seguendo le istruzioni riportate sul manuale, PRIMA di dare Run.

8 Dopo aver fatto partire un programma, in caso, ad esempio, di segnalazione di errore in linea 350, digita soltanto...

List 350

...e accertati che la linea che appare sul video sia RIGOROSAMENTE identica a quella stampata sulla rivista.

9 Se, digitando un listato, ti accorgi che vi sono istruzioni di tipo Poke e Sys, raddoppia la prudenza nella digitazione e nelle procedure di registrazione PRIMA di dare Run.

10 Accertati che il programma che ti accingi a digitare sia REALMENTE valido per il tuo computer. Su ogni articolo della nostra rivista (da leggere SEMPRE con attenzione) è indicato il tipo di computer per il quale il listato stesso è idoneo.

```
100 REM MATEMATICA UTILE CON IL C/64
110 REM SOMMATORIA E FORMULE INVERSE
120 :
130 REM A+B+C+D+E+F=G  (FORMULA PRESA IN CONSIDERAZIONE)
140 FOR I=1 TO 11: X$=X$+CHR$(32):NEXT:X$=CHR$(157) + X$
150 PRINT CHR$(147);:REM CANCELLA SCHERMO
160 PRINT "A+B+C+D+E+F=G":REM VISUALIZZA FORMULA
170 PRINT CHR$(19):PRINT:FOR I= 1 TO 7
180 PRINT CHR$(64+I)"=":NEXT
190 PRINT CHR$(19):PRINT:FOR I= 1 TO 7
200 INPUT"■■■";A$(I):NEXT:REM CURSORE A SINISTRA
210 X=0:FOR I=1 TO 7:REM INDIVIDUA INCOGNITA
220 IF A$(I)="X" THEN Y=I:X=I:I=7
230 NEXT
240 IF X=0 THEN PRINT:PRINT"ERRORE "X$ :GOTO170
250 W=0:FOR I=1 TO 6:W=W+VAL(A$(I)):NEXT
260 W=W-VAL(A$(7)):REM SE INCOGNITA E' G...
270 IF X<7 THEN W=-(W-VAL(A$(X))):REM ...SE NON LO E'
280 PRINT:PRINT CHR$(64+X)"= (X)= "W;X$:GOTO 170
290 END
```

```
100 REM MATEMATICA UTILE CON IL C/64
110 REM PERCENTUALI E FORMULE INVERSE
120 :
130 REM QF=QI*(1+P/100) FORMULA INIZIALE
140 FOR I=1 TO 11: X$=X$+CHR$(32):NEXT
150 Y$=X$+X$: X$=CHR$(157) + X$
160 PRINT CHR$(147);:REM CANCELLA SCHERMO
170 PRINT "QF=QI*(1+P/100)"
180 PRINT CHR$(19): PRINT: PRINT
190 PRINT "QF=";: INPUT QF$: QF=VAL(QF$)
200 PRINT "QI=";: INPUT QI$: QI=VAL(QI$)
210 PRINT "P =";: INPUT P$ : P =VAL( P$)
220 X=0:IF QF$="X" THEN X=X+1
230 IF QI$ = "X" THEN X=X+1
240 IF P$  = "X" THEN X=X+1
250 IF X=0 OR X>1 THEN PRINT " ERRORE":GOTO 180
260 IF QF$="X" THEN GOSUB 300: GOTO 180
270 IF QI$="X" THEN GOSUB 310: GOTO 180
280 IF P $="X" THEN GOSUB 320: GOTO 180
290 GOTO180
300 QF=QI*(1+P/100):PRINT Y$:PRINT QF;X$:RETURN
310 QI=QF/(1+P/100):PRINT Y$:PRINT QI;X$:RETURN
320 IF QI=0 THEN PRINT"ERRORE (QI=0)":RETURN
330 P=((QF/QI)-1)*100:PRINT Y$:PRINT P;X$:RETURN
340 END
```

VUOI COLLABORARE CON COMMODORE COMPUTER CLUB?

Molti lettori desiderano proporre articoli e programmi da pubblicare sulla nostra rivista. Per consentire l'esame del materiale proposto, ma anche per evitare spiacevoli malintesi, riportiamo qui di seguito le norme da seguire per veder pubblicato (e compensato!) un proprio lavoro.

• NON inviare programmi e/o articoli che non siano stati espressamente concordati con la Redazione.
• Per risolvere il problema precedente, telefonare in Redazione (02 / 84.67.34.8) per proporre la collaborazione.
• Si tenga ben presente che non potranno essere accettati programmi che, pur rispondenti al tema concordato, siano privi di corrispondente articolo esplicativo.
• Una volta concordato il "tema", e portato a termine articolo ed eventuale programma, inviarli esclusivamente su FLOPPY DISK; tale necessità deriva dal fatto che, per questioni di tempo, non ci è possibile digitare nè l'uno nè l'altro. E' facoltativo l'invio del lavoro riprodotto anche su carta.
• Per redigere l'articolo, servirsi di un Word Processor commercializzato: Easy Script C/64, Superscript C/128, Word Pro 3, Wordcraft Vic 20, W/P del Plus/4, Magic Desk o altri concordati nel corso della suddetta telefonata. Nel caso di Amiga usare solo file di "Notepad" ed evitare dischetti in auto-boot. Sono da escludere W/P ideati e scritti dall'autore, anche se allegati sul disco inviato.
• I lavori verranno compensati solo ad avvenuta pubblicazione ed in proporzione alla qualità, interesse e rispondenza a quanto concordato; la cifra minima è fissata in L. 25000 per pagina pubblicata. Esempio: articolo e programma che occupano 4 pagine della rivista: compenso minimo L. 100 mila.
• Gli articoli che non dovessero esser pubblicati (a causa della mancata rispondenza a quanto concordato, o per altri motivi) non verranno compensati.
• La Systems Editoriale si riserva il diritto di riportare il materiale inviato, in tutto o in parte, sulle altre testate.
• I lavori inviati allo scopo di comparire soltanto su "Directory", oppure "Amigazzetta", saranno compensati con materiale prodotto dalla Systems Editoriale che si impegna, altresì, a mantenere in evidenza il nome dell'autore nei file inviati.
• Il materiale inviato non viene restituito al mittente in nessun caso.
• Gli articoli e/o programmi inviati sono da considerare a totale disposizione della Systems Editoriale, fatte salve le norme precedentemente esposte.

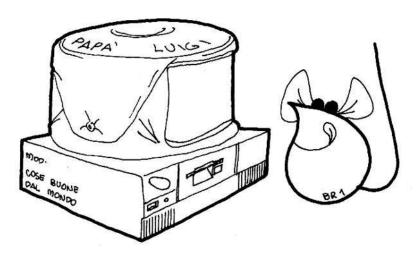
Rispettando le disposizioni di cui sopra eviterete perdite di tempo, di denaro (per spedizioni non richieste) e, soprattutto, di pazienza.

ABBONATI ALL'AVVENTURA

AMBROGIO FOGAR T'ASPETTA OGNI MESE IN EDICOLA

COMPRIAMOCI UN GELATO

Durante l'estate non c'è nulla che sia più gustoso di un bel gelato; se, però, piove o la città è deserta...

Il breve programma in Basic affronta un problema classico

Una delle più attività più antiche e affascinanti richieste ad un personal computer è sicuramente la possibilità di effettuare simulazioni.

Una tipologia di problemi piuttosto nota è, infatti, quella che gli americani chiamano *What if* (che cosa accadrebbe se...) e si riferisce alla determinazione di particolari situazioni il cui esito dipende da fattori concomitanti.

Un tipico problema *Whath if* potrebbe essere il seguente: che cosa accade se un'automobile, che trasporta cinque passeggeri, affronta una curva di 30 gradi in salita alla velocità di 70 Km/h durante un temporale misurabile in 65 mm di pioggia all'ora?

Una simile domanda è certamente legata ad una sana curiosità: sbanda o non sbanda? Ovviamente potrebbero intervenire altri parametri nella determinazione di una corretta risposta: stato di usura delle gomme, tipo di pavimentazione, marcia innestata al momento della curva, tipo di trazione e così via. Più numerosi saranno i dettagli, più precisa sarà la risposta.

DALLE AUTO AI GELATI

Un classico problema informatico, riportato spesso quasi per burla dagli esperti, è quello del gelataio: sì, avete capito bene.

Un gelataio ambulante decide di stabilire il suo chiosco mobile in una certa località: quanti ingredienti deve preparare (e quindi acquistare) per fare in modo di tornare a casa, al termine della giornata, senza un grammo di gelato ed essendo sicuro che non avrebbe venduto altri gelati?

Se, infatti, preparerà un numero di gelati superiore alla effettiva richiesta (caso di una giornata di pioggia, periodo non feriale, località poco frequentata) gran parte del prodotto resterà invenduta e costitui-

LE AVVENTURE DI

PRIMO GIOVEDINI

by Marco Mietta
ha collaborato
Barbara De Toffoli

Buon compleanno, Primo!

La casualità delle condizioni è posta alla base del gioco

rà quindi una *perdita* (il gelato è un bene deperibile e non può esser conservato a lungo).

Se la quantità di gelato risulta inferiore alle effettive richieste (giornata particolarmente calda, periodo di vacanza, località affollata) il gelataio venderebbe, sì, interamente il suo prodotto, ma l'impossibilità di evadere altre richieste costituirebbe un *mancato guadagno*.

L'ideale, quindi, è indovinare con la massima precisione la quantità esatta di gelato da produrre, in modo da venderlo tutto.

In un prossimo futuro vi saranno reti di calcolatori in grado di prevedere esattamente la situazione meteorologica, i "flussi" di persone che si dirigono verso ben precise località, il numero di concorrenti (altri gelatai ambulanti) che intendono recarsi in tali località ed altri dati di notevole interesse per la soluzione del problema.

Quando il progresso tecnologico consentirà, ad un comune gelataio, di effettuare facilmente i suoi calcoli (e con il minimo di spesa), solo allora potremo affermare di vivere in una società *informatica*. Contemporaneamente avremo risolto il terribile problema sociale (tipicamente occidentale) degli *sprechi*.

Nel frattempo, il gelataio deve fidarsi della sua esperienza quotidiana...

IL PROBLEMA

Il problema, supponendo di conoscere tutti i dati a priori, è di facile soluzione ed implementabilità; il che significa che si può adoperare un algoritmo sufficientemente credibile in modo da trasferirlo, in seguito, su un calcolatore.

Supporremo, quindi, che il nostro gelataio dispone, all'inizio, di un capitale di 100 mila lire. Supporremo inoltre che, per vivere, necessita di 10 mila lire al giorno (immaginiamo di avere a che fare con un gelataio dalle esigenze piuttosto contenute). Ogni giorno egli acquista una certa quantità di ingredienti con i quali prepara i gelati che venderà, l'indomani, in una certa località.

Vi sono alcuni parametri che influiranno sulla (mancata) vendita dei gelati:

La località può esser molto frequentata (maggiori probabilità di vendita, ricavi maggiori delle spese), normalmente frequentata (ricavi eguali alle spese) e poco frequentata (ricavi minori delle spese).

Analogamente la qualità degli ingredienti può essere elevata (maggior possibilità di attirare l'interesse del pubblico), normale o scadente; le vendite subiranno un andamento del tutto simile al caso precedente.

La giornata può appartenere ad un periodo di vacanza, festivo o feriale e la sua temperatura (determinante per invogliare all'acquisto di un gelato) può essere elevata, normale o bassa.

Per ognuno dei quattro parametri (temperatura, localtà, periodo, qualità) vi sono quindi tre possibilità di vendita. Alla prima corrisponde un guadagno del 10%, alla seconda un guadagno nullo ed alla terza una perdita del 10%

E' ovvio che le percentuali si incrementano o si elidono a seconda dei casi. Inutile dire, infine, che bisogna guadagnare, giorno per giorno, quanto più è possibile tenendo conto che, anche in caso di inattività, il gelataio spende, per vivere, ben 10 mila lire al giorno.

IL PROGRAMMA

Un programma *What if*, in grado di affrontare il problema del gelataio, può essere implementato in vari modi diversi; noi ci riferiremo alla realizzazione di un gioco sufficientemente divertente(?).

Sarà, in ogni caso, necessario usare intensamente la funzione RND che, generando numeri casuali, è in grado di creare artificialmente numerose situazioni diverse. Limitandosi, però, a far apparire una serie di precisazioni...

La giornata è calda
Il luogo non è affollato
Il periodo è di vacanza
La qualità è ottima

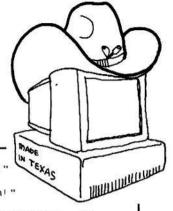
...seguite dalla fatidica domanda...

Quanti gelati prepari?

...non darebbe soddisfazione dal momento che è piuttosto agevole trovare la risposta. In questo caso dovrebbe esser necessario impostare un cronometro in modo da evidenziare, da parte del giocatore, le sue doti di velocità di interpretazione dei dati.

Un altro modo di impostare il gioco potrebbe essere quello di formulare dapprima la domanda, far

Luglio 1989: eh, sì, amici, siamo un'altra volta a Luglio, dopo 12 mesi passati in compagnia del nostro Primo Giovedini! Giusto un anno fa, infatti, con l'episodio "Galaxian Invasion", aveva inizio la serie ...

... di avventure di questo personaggio, che ha accompagnato i lettori di C.C.C. fino ad oggi. Durante tutto questo tempo Primo Giovedini si è cimentato in imprese eroiche, ai limiti del formattabile, sfruttando ogni più piccola RAM disponibile ...

```
100 PRINT CHR$(147) "COMPRIAMOCI UN GELATO!"
110 PRINT"MINI-PROGRAMMA DISIMPEGNATO"
120 PRINT"PER GENTE PIGRA, MA NON SFATICATA!"
130 :
140 CAPITALE=100000:PRINT:PRINT"(ATTENDI UN MOMENTO...)"
150 PRINT: FOR I=1 TO 4:FOR J=1TO3
160 READ CO$(I,J):NEXT J:NEXT I
170 FOR I=1 TO 4
180 GOSUB 470
190 IF DO$(I,X)="" THEN READ DO$(I,X)
200 A=0:FOR B=1 TO 3:IF DO$(I,B)<>"" THENA=A+1
210 NEXT B:IF A<3 THEN 180
220 NEXTI
230 GOSUB470:PRINT DO$(1,X):Y=(2-X)/10
240 GOSUB470:PRINT DO$(2,X):Z=(2-X)/10
250 GOSUB470:PRINT DO$(3,X):W=(2-X)/10
260 GOSUB470:PRINT DO$(3,X):W=(2-X)/10
270 PRINT:INPUT"QUANTI INGREDIENTI ACQUISTI";QI
280 IF QI>CA OR QI<-CA THEN PRINT"NON LI HAI!":GOTO270
290 CA=CA-QI-10000
300 QI=QI*(1+Y)*(1+Z)*(1+W)*(1+K):CA=INT(CA+QI)
310 IFCA<0THEN PRINT:PRINT"HAI PERSO!":PRINT:GOTO480
320 PRINT CA:GOTO230
330 GET A$:IF A$="" THEN 330
340 RETURN
350 :
360 REM NOMI IN "CHIARO"
370 DATA CALDO, NORMALE, FREDDO
380 DATA ZONA FREQUENTATA,ZONA NORMALE,ZONA DESERTA
390 DATA PERIODO DI VACANZA, PERIODO FESTIVO,PERIODO FERIALE
400 DATA QUALITA' OTTIMA,QUALITA' NORMALE,QUALITA' SCADENTE
410 :
420 REM NOMI IN "CODICE"
430 DATA TEMPO 1, TEMPO 2, TEMPO 3
440 DATA ZONA 1,ZONA 2,ZONA 3
450 DATA PERIODO 1, PERIODO 2,PERIODO 3
460 DATA QUALITA' 1,QUALITA' 2,QUALITA' 3
470 X=INT(1+RND(1)*3):RETURN
480 PRINT:PRINT"ECCO I NOMI IN CODICE USATI...":PRINT
490 FOR I=1 TO 4:FOR J=1 TO 3
500 PRINT CO$(I,J)CHR$(18)DO$(I,J):NEXT:PRINT:NEXT
510 END
```

...il giocatore non saprà, ad esempio, se a *Qualità 3* corrisponde una qualità ottima, normale o scadente.

Dopo vari tentativi, e tenendo conto delle elaborazioni del computer, il giocatore potrà, tuttavia, stabilire le effettive corrispondenze tra i nomi in codice ed in chiaro.

Il gioco termina quando il gelataio rimane senza soldi. A questo punto vengono visualizzate le corrispondenze tra i nomi in chiaro e quelli cifrati.

Un ultimo particolare: alla domanda che compare è necessario rispondere digitando la somma (in lire) che si desidera investire per l'acquisto degli ingredienti. Ricordiamo che anche rispondendo con zero (0) verrà detratta, dal capitale a disposizione, la somma di 10 mila lire al giorno. Indicando, invece, una cifra negativa, si indicherà, al computer, l'intenzione di rivendere al grossista una certa quantità di ingredienti. Tale ultima opportunità potrebbe rivelarsi utile nel caso in cui, il giorno successivo, la situazione prevista sia giudicata del tutto sfavorevole alla vendita di gelati.

Tentate di individuare il piccolo errore presente nel listato

apparire, in seguito, la *situazione globale* e trarre le dovute conclusioni.

In questo modo di operare, tuttavia, la fortuna e la casualità rivestirebbero un ruolo determinante ed il gioco avrebbe lo stesso senso del tiro dei dadi.

Il listato di queste pagine costringe il giocatore, invece, a ragionare prima di indicare le sue scelte.

All'inizio, infatti, alla matrice CO$ vendono associati i 12 messaggi relativi alle tre situazioni possibili per ciscuno dei quattro parametri (righe DATA "Nomi in chiaro").

Subito dopo, nella matrice "parallela" DO$, vengono allocati, in modo casuale, altrettanti messaggi dal nome generico Zona 1, Zona 2, ... Qualità 1, Qualità 3 la cui corrispondenza con i relativi elementi di CO$ perdurerà per l'intera durata del gioco.

Quando su video compare, pertanto, una situazione di questo tipo...

Tempo 2
Zona 2
Periodo 1
Qualità 3

Quanti gelati prepari?

SOTTO L'OMBRELLONE

Il periodo estivo, si sa, è degno di un vigoroso, confortevole e disarmante dolce far niente. Se state leggendo queste righe sulla spiaggia, o in pineta, l'occasione si rivela propizia per saltare a piè pari le pagine dedicate al listato.

Per coloro che non hanno troppo caldo, (o che desiderano verificare fino a che punto la propria materia grigia perda colpi sotto il sole cocente) vi proponiamo una mini-micro-sfida che non richiede la presenza del calcolatore (il suo uso, anzi, è rigorosamente vietato).

Il listato contiene, volutamente, un piccolo errore di programmazione: a voi tocca individuarlo e correggerlo adeguatamente. Per aiutarvi diremo solo che sarà importante la qualità della ricerca che effettuerete...

Inoltre, se avete a portata di mano una penna o una matita, provate a stendere un algoritmo che risulti più efficiente di quello che, presente da riga 190 a 220, è incaricato di assegnare i nomi in codice alla matrice CO$. Per noi l'algoritmo pubblicato è il migliore possibile. E per voi?

Arrivederci a settembre e, mi raccomando, non sforzatevi troppo (poco)!

A RIGOR DI LOGICA

Per capire come "ragiona" un computer è indispensabile esaminare i principi elementari sui quali si basano i chip al silicio

di **Alessandro de Simone**

Per funzione logica si intende quella particolare funzione che, basandosi su alcuni dati presenti in *ingresso*, impone un ben definito stato logico in *uscita*.

Tra le figure di queste pagine notiamo le cosiddette tavole della verità che possono essere considerate le *tabelline* di scolastica memoria.

La funzione (nota anche come *porta*) AND, ad esempio, presenta lo stato logico uno (1) in uscita sole se *entrambi* gli ingressi sono posti a uno. Negli altri tre casi (cioè coppia di valori 00, 01, 10) il risultato è sempre zero.

Molto spesso capita, in effetti, di avere a che fare con chip (circuiti integrati) che presentano molti ingressi anziché due soltanto. Vi sono anche alcuni chip che, oltre a presentare un'uscita "ufficiale", dispongono di altri pin (piedini) sui quali sono presenti i risultati di altrettante elaborazioni svolte basandosi, ancora una volta, sui dati presenti in ingresso. Ci riferiamo, tra l'altro, all'eventuale carry (riporto) e all'uscita complementare (che rappresenta, in pratica, il contrario dell'uscita standard). Vi possono anche essere numerose uscite, come nel caso in cui è necessario comandare i singoli segmenti di un display luminescente.

In futuro ci soffermeremo su questi particolari circuiti integrati, dal prezzo irrisorio, e proporremo esperimenti harware e software. In questa sede ci limiteremo a simulare, con il computer, le quattro funzioni logiche più note (AND, OR, NOR, NAND) operando sullo schermo del C/64.

LA SIMULAZIONE

Dopo aver impartito il solito Run compariranno, sul video, tre rattangolini in reverse, disposti a triangolo. All'inizio essi rappresenteranno tre funzioni di tipo And; ai loro ingressi saranno presenti, inoltre tre coppie di zeri.

Mediante i tasti *1, 2, 3* sarà possibile modificare la funzione di ciascun chip; grazie alla pressione dei tasti *W, E*, invece, sarà possibile modificare la coppia di bit presenti alla base dei due chip inferiori. I bit in uscita di questi ultimi, infatti, rappresentano l'ingresso del terzo chip.

Le combinazioni possibili sono piuttosto numerose: si tratta di quattro possibili ingressi (00, 01, 10, 11) applicati a due chip che possono contenere, ciascuno, una delle quattro funzioni logiche *And, Or, And, Nand*. Il terzo chip, infine, offre anch'esso una scelta su quattro funzioni diverse. Provate a calcolare il numero di combinazioni possibili.

IL PROGRAMMA

Il programma, che funziona perfettamente, forse non rappresenta il massimo dell'efficienza: permette, è vero, un agevole inserimento di altre funzioni logiche (*Or eclusivo, Nor esclusivo*) ma presenta alcuni ostacoli nel caso si desiderasse inserire ampliamenti di un certo respiro.

Inutile dire che, anche in questo caso, chiediamo al lettore di intervenire opportunamente, studiando la possibilità di connettere tra loro più funzioni logiche (magari dotate di tre ingressi, anziché due soltanto) e collegate tra loro nel modo più disparato.

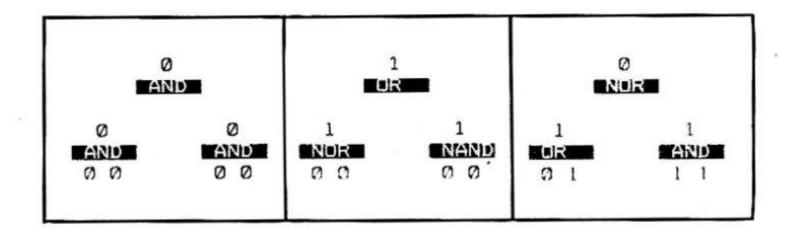
Anche per ciò che riguarda la velocità il programma può lasciare a desiderare. Dal momento che esso opera in tempo reale (l'intero circuito modifica immediatamente il proprio stato dopo ciascuna variazione in ingresso) l'elaborazione richiede un certo tempo, breve con il listato proposto, che può risultare inaccettabile inserendo altri circuiti, prevedendo altri ingressi e, in definitiva, aumentando il

Tutti i computer basano il proprio funzionamento sulle porte logiche

Il listato di queste pagine simula l'azione di tre circuiti integrati

numero di elaborazioni (e di visualizzazioni) che il computer è costretto a compiere.

La soluzione ideale può esser costituita dalla concatenazione di subroutine che (come nelle righe da 1950 a 2500) consente l'inserimento di un numero pressochè illimitato di funzioni ed ingressi selezionabili (vedi anche le righe da 2550 a 2950).

Per ciò che riguarda la digitazione, questa non presenta particolari problemi a patto di prestare la massima attenzione anche ai particolari più insignificanti. Ci riferiamo, in particolare, alla presenza del carattere virgola (,) nelle righe 700 e 800 (che impone il corretto incolonnamento dei dati) e al conteggio preciso degli spazi, ove richiesto. Questi sono in numero di cinque nelle righe 900 e 850; i caratteri speciali presenti, prima dei cinque spazi, in riga 850, sono dei *crsr up* (cursore in alto).

PRIMA DI SALDARE

Di recente abbiamo pubblicato un articolo che descrive la programmazione della porta utente del C/64.

I listati che lo corredavano si limitavano ad accendere un banalissimo Led (mini-lampadina allo stato solido) ed a "riconoscere" la pressione di un pulsante esterno.

Per ampliare le possibilità offerte dalla porta utente è necessario impiegare altri componenti che consentano di parlare un linguaggio riconoscibile da altre apparecchiature, digitali anch'esse, che basano il loro funzionamento sull'elaborazione di due soli dati: lo zero (0) e l'uno (1).

In questo articolo non torneremo a ripetere argomenti arcinoti, ma richiameremo la vostra attenzione sulle tavole della verità e sull'elaborazione che i più comuni circuiti al silicio sono in grado di espletare.

Completerà la chiacchierata l'immancabile programma in Basic (valido per qualsiasi computer) che consentirà ai meno esperti di comprendere a fondo la logica sulla quale un qualsiasi computer (passato, presente e futuro) basa i propri ragionamenti.

Chi, quindi, non ha mai preso un saldatore in mano, e brucia dalla voglia di costruire apparecchietti da collegare al computer, è bene che legga con la massima attenzione le note di queste pagine.

In un prossimo futuro, infatti, abbiamo intenzione di...

Porta AND

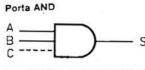
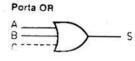
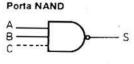
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Porta NOR

A	B	S
0	0	1
1	0	0
0	1	0
1	1	0

Porta OR

A	B	S
0	0	0
1	0	1
0	1	1
1	1	1

Porta NAND

A	B	S
0	0	1
1	0	1
0	1.	1
1	1	0

APRIAMO LA PORTA

Non tutti sanno che cosa siano le porte logiche nè, tantomeno, le *tavole della verità* di cui si parla nell'articolo.

E' bene, quindi, soffermarsi sui principali significati di questi termini.

Per tavola della verità si intende una griglia in cui, per ciascuna configurazione di ingresso, è presente l'uscita corrispondente. Dal momento che gli *stati di ingresso* possono essere soltanto *1* oppure *0*, ne consegue che sono possibili tante combinazioni quante ne rappresenta, appunto, il numero di stati possibili (cioè, due) elevato al numero di ingressi disponibili. Se, ad esempio, si hanno a disposizione tre ingressi, il numero di combinazioni possibile sarà 2 elevato a 3, cioè 8: *000, 001, 010, 011, 100, 101, 110, 111*.

Nelle figure di queste pagine, per ciascuna porta sono indicati due soli ingressi, mentre un terzo (eventuale) ingresso è tratteggiato. Per ciascuna porta è anche riportata la corrispondente tavola della verità il cui significato è di immediata comprensione:

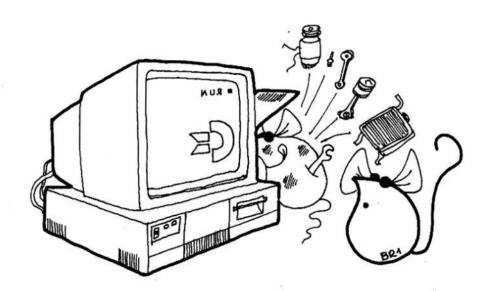
Una porta di tipo *OR* (che in inglese vuol dire *oppure*) presenta un "uno" in uscita se *almeno* uno dei suoi ingressi possiede un "uno".

Una porta di tipo *AND* (che in inglese vuol dire *anche*) presenta un "uno" in uscita se *entrambi* i suoi ingressi possiedono un "uno".

Una porta di tipo *NAND* (abbreviazione inglese di Not And) presenta uno "zero" in uscita se *entrambi* i suoi ingressi possiedono un "uno".

Una porta di tipo *NOR* (abbreviazione inglese di Not Or) presenta un "uno" in uscita se *entrambi* i suoi ingressi possiedono uno "zero".

Negli schemi elettronici le varie porte sono individuabili dalle caratteristiche forme loro assegnate. Si noti, infatti, la presenza di un pallino che distingue la porta Nand da quella And. Analogamente si noti la parte inferiore della porta Or (di tipo curvo) che la contraddistingue da quella di tipo And.

```
100 PRINT CHR$(147) "ESPERIMENTI SUI CIRCUITI LOGICI"
150 PRINT:PRINT "CON I TASTI 1, 2, 3 MODIFICHI I CIRCUITI"
200 PRINT "CON I TASTI W, E MODIFICHI GLI INGRESSI"
250 :
300 A$(1)=" 0 0 ":A$(2)=" 0 1 ":REM QUATTRO COMBINAZIONI
350 A$(3)=" 1 0 ":A$(4)=" 1 1 ":REM POSSIBILI (INGRESSI)
400 CI$(1)="■ AND ■":CI$(2)="■ OR  ■":REM QUATTRO FUNZIONI
450 CI$(3)="■ NAND■":CI$(4)="■ NOR ■":REM DISPONIBILI
500 C1$=CI$(1):C2$=C1$:C3$=C1$:REM INIZIALIZZAZIONI VARIE
550 J$=A$(1):B2$=J$:B3$=B2$    :REM TRE CIRCUITI AND
600 X=1:Y=1:Z=1:XW=1:XE=1      :REM INGRESSI POSTI A 0
650 PRINT CHR$(19):FOR I=1 TO 10: PRINT: NEXT
700 J$="": K$=C2$:H$=B2$:GOSUB2550:PRINTR$,:J$=" "+Z$+" "
750 K$=C3$:H$=B3$:GOSUB 2550:PRINT R$:J$=J$+Z$+" "
800 PRINT C2$, C3$: PRINT B2$, B3$:REM ****************
850 K$=C1$: H$=J$: GOSUB 2550: PRINT"▯▯▯▯▯▯      "R$
900 PRINT "      " C1$: PRINT
950 GOSUB 1800:REM ESAME PRESSIONE TASTO
1000 IF A$="1" THEN GOSUB 1300: GOTO 650 :REM FUNZ. ALTO
1050 IF A$="2" THEN GOSUB 1400: GOTO 650 :REM FUNZ. SINISTRA
1100 IF A$="3" THEN GOSUB 1500: GOTO 650 :REM FUNZ. DESTRA
1150 IF A$="W" THEN GOSUB 1600: GOTO 650 :REM INGR. SINISTRA
1200 IF A$="E" THEN GOSUB 1700: GOTO 650 :REM INGR. DESTRA
1250 GOTO 950:REM IGNORA ALTRI TASTI
1300 X=X+1:IF X>4 THEN X=1      :REM SELEZIONE DELLA FUNZIONE
1350 C1$=CI$(X):RETURN          :REM DEI TRE CHIP.
1400 Y=Y+1:IF Y>4 THEN Y=1      :REM E' POSSIBILE SCEGLIERE
1450 C2$=CI$(Y):RETURN          :REM TRA CIRCUITI DI TIPO
1500 Z=Z+1:IF Z>4 THEN Z=1      :REM AND, OR, NAND, NOR
1550 C3$=CI$(Z):RETURN
1600 XW=XW+1:IF XW>4THEN XW=1   :REM SELEZIONE DEGLI INGRESSI
1650 B2$=A$(XW):RETURN          :REM E' POSSIBILE SCEGLIERE
1700 XE=XE+1:IF XE>4THEN XE=1   :REM UNA DELLE QUATTRO COMBIN.
1750 B3$=A$(XE):RETURN          :REM POSSIBILI: 00, 01, 10, 11
1800 GET A$:IF A$="" THEN 1800
1850 RETURN
1900 :
1950 Z$="0": REM CIRCUITO AND          :REM DETERMINA
```

```
2000 IF X$="1" AND Y$="1" THEN Z$="1"  :REM IL RISULTATO
2050 RETURN                             :REM DELLE FUNZIONI
2100 Z$="0": REM CIRCUITO OR
2150 IF X$="1" OR Y$="1" THEN Z$="1"   :REM ALL'USCITA
2200 RETURN                             :REM DELLA SUBROUTINE
2250 Z$="1": REM CIRCUITO NAND          :REM LA VARIABILE Z$
2300 IF X$="1" AND Y$="1" THEN Z$="0"  :REM CONTIENE
2350 RETURN                             :REM UNO DEI DUE
2400 Z$="0": REM CIRCUITO NOR           :REM POSSIBILI VALORI
2450 IF X$="0" AND Y$="0" THEN Z$="1"  :REM CIOE' 1 OPPURE 0
2500 RETURN
2550 IF H$=A$(1) THEN X$="0":Y$="0": REM ASSEGNA I VALORI
2600 IF H$=A$(2) THEN X$="0":Y$="1": REM DI INGRESSO
2650 IF H$=A$(3) THEN X$="1":Y$="0": REM A SECONDA
2700 IF H$=A$(4) THEN X$="1":Y$="1": REM DELLA COMBINAZIONE
2750 IF K$=CI$(1) THEN GOSUB 1950: REM INVIO ALLA SUBROUTINE
2800 IF K$=CI$(2) THEN GOSUB 2100: REM CHE ELABORA
2850 IF K$=CI$(3) THEN GOSUB 2250: REM LA FUNZIONE LOGICA
2900 IF K$=CI$(4) THEN GOSUB 2400: REM SELEZIONATA
2950 R$= "  "+Z$:RETURN:REM CONCATENAZIONE PER OUTPUT OTTIMALE
3000 END
```

CENTRIAMO IL BERSAGLIO, MA CON BRIO

Poche righe in Basic sono sufficienti per realizzare un divertente gioco di percorso; e per imparar qualcosa

di **Gregor Samsa**

Per digitare il breve listato bastano venti minuti circa

"Al cuore, Ramon...!"

Gli appassionati di film western, anzi spaghetti - western, avranno certo riconosciuto la citazione (non proprio dotta), tolta di bocca ad un duro per eccellenza, Clint Eastwood.

Per l'occasione il nostro eroe invitava il cattivo (che più cattivo non si può) di turno a mirare, dritto e senza esitazioni, contro un bersaglio decisamente... organico.

Niente paura, non avete sbagliato rivista: siete sempre sintonizzati sulle frequenze di Commodore Computer Club, e la testata non ha improvvisamente dirottato i suoi interessi verso la critica cinematografica.

Se vi state chiedendo quale rapporto può intercorrere tra un duello a suon di Colt ed il nostro inoffensivo mouse di Amiga, ebbene è presto detto: di antitesi (ora anche la filosofia!).

Nel senso che, per raggiungere un bersaglio, non è detto che si debba obbligatoriamente percorrere la via più diretta: anzi, può risultare molto più divertente "complicare" il percorso ideale, creando un passatempo di sicura efficacia.

Videogames tipo Gun Shoot non sono poi da buttar via, ma anche noi, con un più modesto basic, possiamo realizzare qualcosa di simpatico, e non solo fine a se stesso: purchè lo si voglia, c'e sempre da imparare.

Più in particolare vedremo come sia possibile, anche senza ricorrere a *Bob* e *Sprite*, gestire un minimo di movimento sullo schermo, e come far interagire grafica e testo.

Il tutto, ovviamente, in modo abbastanza semplice, anche se con i limiti di velocità imposti dal basic.

Limiti che, per quanto aggirabili, nel nostro caso possono tuttavia andar bene, date le caratteristiche del gioco.

DESTREZZA E STRATEGIA

Vediamo, dunque, di cosa si tratta, cominciando con il copiare il listato di queste pagine dopo aver avviato l'esecuzione di AmigaBasic cliccando sulla corrispondente icona.

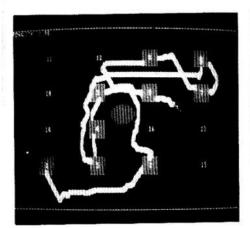
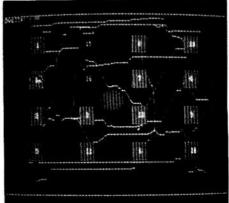
Si tenga presente che i commenti, preceduti dal singolo apice (apostrofo, o accento che dir si voglia), non sono strettamente indispensabili; tuttavia una certa chiarezza non guasta mai: potreste decidere, magari tra qualche mese, di apportare al programma delle modifiche sostanziali, e "ristudiare" un listato è sempre più noioso che digitare qualche riga in più.

Stesso discorso, come ovvio, per righe vuote e indentazioni (allineamenti) varie.

Ultimata l'operazione, si salvi il programma, quindi lo si mandi in esecuzione con Run in modo diretto, o selezionando Start dal menu *Run*.

A questo punto, se tutto è in regola (= non si sono commessi errori di copiatura), apparirà una richiesta di input riguardante il livello di difficoltà.

Verrà esaminato, tra breve, cosa comporta tale scelta: per ora ci si limiti (per esempio) a digitare *1* e Return.

Ed ecco la nostra schermata di lavoro(!), compo-

[illegible] riquadri neri, numerati casualmente (ma sempre da 1 a 16), ed un cerchio viola al centro: il bersaglio da colpire.

I colori descritti risulteranno tali solo se non si è modificata la gamma standard attraverso le *Preferences*: anche in questo caso, tuttavia, nonostante l'estrema importanza dei colori nella gestione del programma, non sorgerà alcun problema.

Scopo del gioco è raggiungere il bersaglio con il minor numero di penalità possibili, dopo aver toccato, in rigorosa sequenza numerica, tutti i riquadri neri.

Troppo facile?

Aspettate di conoscere il resto.

Si muova, quindi, il puntatore del mouse fino a portarlo in prossimità (o, se si è già pratici, all'interno) del riquadro 1, e si prema il pulsante sinistro del mouse una sola volta (durante il gioco non va tenuto pressato).

Da questo momento, ogni spostamento del topo produrrà il tracciamento di una linea continua, che si formerà più lentamente rispetto allo spostamento del *pointer* (= freccetta del mouse).

Vedremo più avanti come si possa sveltire (se è il caso) le operazioni, ma, data la precisione richiesta dal gioco, si consiglia addirittura di modificare (al valore massimo) la velocità di escursione del mouse attraverso le Preferences.

Come prima regola, la scia bianca lasciata dal mouse non può essere più toccata, se non assommando delle penalità, il cui ammontare viene visualizzato in alto a sinistra dello schermo.

Se ora ci si porta all'interno del riquadro 1 con mossa decisa (solo con la pratica si capirà il modo migliore di accesso senza pagare alcuna penalità) questo si *accenderà* (cambiando il proprio colore) e da questo momento in poi potrà essere attraversato impunemente.

Ora non resta che "incamminarsi" verso il riquadro 2 (e così via fino al 16), tenendo presente che è vietato passare in riquadri ancora spenti (= neri).

Facendolo, si assumeranno parecchie penalità, la cui entità dipende dal numero di pixel attraversati, nonchè dallo spessore della stessa scia tracciata sullo schermo.

Passando nelle zone *vietate*, infatti, la traccia del mouse non viene visualizzata, e può capitare anche di dover attendere qualcosina in più per vederla riapparire al di là dell'ostacolo.

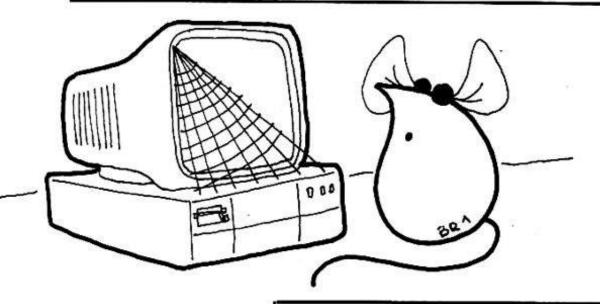
Si badi che anche un minimo spostamento inverso del mouse, o angolato senza attenzione, incrementerà il punteggio, in quanto si tornerebbe involontariamente sulla linea bianca.

Stesso discorso per l'interno dei riquadri: anche se accesi, il numeretto bianco va evitato (difficoltà maggiore, in quanto la linea non risulta visibile).

Il bersaglio, se attraversato prima che si sia raggiunto il numero 16, costituisce un'altra fonte di penalità.

Il raggiungimento dell'ultimo numero segnerà la fine del gioco.

Come si sarà capito, il mini-game si basa su due elementi: una strategia di percorso, in modo da non precludere l'accesso ad un numero che si deve an-

Vi consigliamo di esaminare con cura la parte relativa alla gestione delle coordinate

QUALI COORDINATE?

Lo schermo di Amiga, com'è noto, non fa differenza tra modo testo e modo grafico, come invece avveniva nei suoi predecessori ad 8 bit.

Questo significa, in pratica, che tanto una istruzione *Print*, quanto una *Circle* fanno sempre riferimento alla stessa finestra (e quindi schermo) al momento considerata come output.

Il sistema di coordinate adoperato dai due diversi modi grafici è però alquanto diverso.

Mentre, infatti, qualunque carattere-testo va posizionato secondo una suddivisione dello schermo in righe e colonne, le coordinate che riguardano i comandi grafici sono sempre espresse in pixel.

Talvolta, ed è il caso del programma di queste pagine, è necessario far interagire i due modi di esprimere delle coordinate.

Per esempio, riferendosi al listato, i riquadri neri vengono disegnati adoperando normali caratteri testo, cioè degli spazi con al loro interno i caratteri numerici.

D'altra parte il movimento del mouse, e relativo tracciamento della linea, è espresso in pixel; ciò nonostante è necessario, per il gioco, testare se le coordinate della posizione del mouse corrispondono a quelle dei riquadri toccati.

Ecco che, quindi, si rende necessaria una "conversione", per stabilire a quale pixel corrisponde una certa posizione espressa in termini di riga / colonna.

Il problema, comunque, è abbastanza semplice da risolvere.

Ogni carattere, infatti, a meno che non si intervenga sui *Fonts* particolari, è rappresentato da una griglia di 8 x 8 pixel.

Il primo carattere in alto a sinistra occuperà quindi le coordinate-pixel tra *0, 0* (angolo in alto a sinistra del carattere) e *7, 7* (angolo in basso a destra).

Da ciò diventa facile dedurre la formula di conversione, che risulterà essere...

Angolo alto / sin. = (colonna - 1) * 8, (riga - 1) * 8

Angolo bas. / des. = (colonna * 8) -1, (riga * 8) -1

...con la virgola che separa la coordinata X dalla Y.

Nel listato, per determinare l'estremità destra dei riquadri, variabili *X2()* e *Y2()*, deve anche essere considerato il fatto che si è in presenza di 5 caratteri disposti su 3 righe.

La procedura opposta, per determinare riga e colonna conoscendone il pixel di posizione, sarà invece:

colonna = INT (coordinata X / 8) +1
riga = INT (coordinata Y / 8) + 1

cora raggiungere, nonchè una certa destrezza nel l'uso del mouse (ottimo esercizio!), data la struttura a labirinto che si viene a formare con il procedere degli spostamenti.

Lo spessore della traccia bianca viene deciso dal livello di difficoltà scelto all'inizio; il livello zero produrrà una linea composta da un singolo pixel.

Si consiglia di muovere il mouse con una certa dolcezza, assecondando il graduale formarsi della scia.

Un ultimo avvertimento: se ci si ferma in un punto proibito, le penalità continueranno ad aumentare senza tregua, finchè non ci si sarà tolti dall'impiccio.

Non resta che darsi da fare, senza montarsi la testa: è pressochè impossibile totalizzare 0 penalità.

A bersaglio centrato, un eventuale nuovo percorso risulterà sempre diverso dal precedente, a meno che non si modifichi una particolare istruzione del listato, come vedremo tra breve.

Questa possibilità, tra l'altro, consente eventuali sfide tra più concorrenti.

Adoperando, infatti, lo stesso "campo di battaglia", l'immutata difficoltà di percorso renderà più evidente la gerarchia di valori (piaciuto il parolone?) tra due punteggi raggiunti.

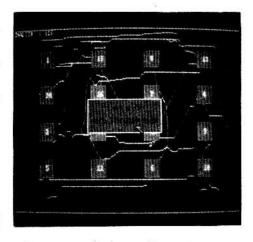
IL LISTATO

Abbiamo accennato, in precedenza, come almeno due caratteristiche del gioco possono essere facilmente modificate: la sequenza numerica che determina il percorso, nonchè la velocità di tracciamento della scia bianca.

Quelle suggerite non sono certo le uniche variazioni possibili, ma la loro pratica applicazione, fornisce l'occasione per un esame dei punti più salienti del programma proposto.

Procediamo con ordine, seguendo lo sviluppo del listato.

Come si può notare, prima di ogni altra cosa viene attivata l'istruzione *On Break*, direzionata verso la subroutine che chiude le finestre e lo schermo attivi.

Tale precauzione si rende necessaria per consentire, qualora lo si desideri, di interrompere il programma tramite l'opzione *Stop* del Menu del basic, senza lasciare parte della memoria impegnata dallo schermo 2.

Questo, tramite l'istruzione *Screen*, viene aperto con una risoluzione di 640 x 250 pixel, e con una profondità 3, il che consente di adoperare un totale di 8 colori.

Per i nostri scopi ne serve solo uno oltre i 4 già disponibili normalmente (=palette da 0 a 3), ed ecco quindi la Palette 4, che identifica il colore del bersaglio e dello sfondo nel messaggio finale (viola).

Ultimata la fase di inizializzazione, vengono stampati i 16 riquadri e relativi numeri al loro interno.

Come si può notare, la procedura è molto semplice, in quanto affidata a dei semplici *Print e Locate* (subroutine "quadrati") all'interno di due cicli For ... Next.

Unico punto più complesso, la determinazione delle coordinate in pixel degli estremi di ogni rettangolo.

E' in questa fase, tra l'altro, che viene assegnato il numero che identifica ogni "postazione", mediante l'uso di *Randomize*.

Eliminando l'istruzione *Randomize Timer*, e mantenendo l'assegnamento successivo xx = INT (RND *16 + 1), la sequenza numerica verrà ripetuta tale e quale ad ogni ripresa del gioco.

E' Randomize Timer, infatti, che reinizializza il generatore di numeri casuali, come del resto precisato nel manuale in dotazione alla macchina (non si dimentichi che esiste anche lui!).

Si noti che, per evitare il ripetersi di un valore, il ciclo identificato dalla label *caso* continua finchè non

Avete il compilatore AC Basic Compiler? E' il momento di usarlo...

COMPILARE E' FACILE

E proprio con programmi come quello qui proposto, che il problema della velocità si fa più cocente che mai.
A dispetto di un Amiga ultraveloce, il basic, o meglio il suo interprete, se non supportato da massicci interventi in linguaggio macchina, mostra tutti i suoi limiti, anche se non proprio intollerabili.

D'altra parte, non è certo obbligatorio doversi dedicare a linguaggi ben più complessi come il *C* e l'*Assembly*.

Un compromesso, tuttavia, esiste: la compilazione di un "sorgente" basic.

Per non restare sul vago, diremo che esiste in circolazione, ma rigorosamente piratato e privo di manuale (eventuali smentite da parte di distributori saranno ben accette), un compilatore riservato ad AmigaBasic: *AC Basic Compiler (V 1.3)*.

A quanto pare, tra l'altro, posseduto da molti.

Ebbene, per questi fortunati, il programma di queste pagine è l'ideale per cimentarsi nella prova.

Unico punto cui prestare attenzione è l'istruzione *Run* presente poco prima dell'inizio delle subroutines, che non viene accettata dal compilatore.

Basterà, comunque, modificare la routine *attesa* in fondo al listato, nonchè la riga che precede la label, nel modo seguente:

```
LOCATE 4, 5: PRINT "premi un tasto"
attesa:
a$ = INKEY$
IF a$= "" THEN GOTO attesa
WINDOW CLOSE 3: WINDOW CLOSE 2
SCREEN CLOSE 2
GOTO start
```

Inoltre va modificata l'istruzione...

Window 2, , , 0, 2

...presente nelle primissime righe, che deve diventare...

Window 2, , , ,2

Per procedere alla compilazione occorre salvare il listato in formato Ascii (*Save "nomeprg", a* in modo diretto) nello stesso disco del compilatore, magari eliminando tutti i vari esempi per far spazio; quindi attivare *AC Compiler* così com'è (opzione *compile*), senza settare altri parametri che non lo siano già di default.

Se, dopo la compilazione, si desidera trasportare il compilato (eseguibile direttamente da Workbench) in un altro dischetto, nella directory *L* dello stesso dovrà anche essere copiato il file *BAS.RL*, presente nella directory *L* di *Ac Compiler*.

Tali soluzioni, come si può capire, sono il risultato di sperimentazioni decisamente empiriche a causa della carenza di manualistica: eventuali contributi informativi da parte vostra saranno quindi ben accetti e, se del caso, pubblicati.

Dopo il trattamento, il nostro labirinto - bersaglio assumerà una dignità insospettabile, anche se, data la maggiore velocità, il punteggio scorrerà un po' troppo in fretta, e non certo in nostro favore.

sono stati assegnati i 16 numeri, controllando, ad ogni passaggio, la variabile *Flag(x)*, che viene posta ad 1 per ogni valore già assegnato.

MUOVERSI, SIGNORI!

Eccoci, dunque, alla fase più calda, il cuore del piccolo (ma sincero) game: il tracciamento della linea legata al movimento del mouse, e la determinazione di eventuali collisioni... penalizzanti.

Il primo aspetto, tra l'altro, si ricollega proprio alle eventuali modifiche di cui si è già fatto cenno.

Con il ciclo *Loop1*, si attende la pressione del tasto sinistro del mouse, rivelata, com'è noto, dalla funzione *Mouse(0)*.

In seguito viene usata la variabile fittizia "T" al solo scopo di richiamare la funzione Mouse(0).

Così facendo è poi possibile leggere in *Mouse(1)* e *Mouse(2)* le coordinate orizzontale e verticale del cursore-mouse (si continui a seguire il listato), che vengono assegnate alle variabili "X" ed "Y", con una correzione (-8 e -5) per visualizzare la posizione non proprio "sotto" la freccetta-puntatore.

Il tracciamento vero e proprio viene affidato ad una istruzione *Line*, piuttosto che al classico *Pset*, in modo da avere una maggiore corposità.

In pratica è come se venisse disegnata una serie continua di piccoli riquadri, i cui estremi sono dati da X, Y (angolo in alto a sinistra), e dalle stesse coordinate sommate al valore richiesto in input all'inizio del gioco (per l'angolo inferiore destro).

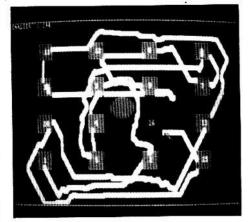
Se quest'ultimo valore è nullo, l'effetto sarà identico a quello che si otterrebbe con Pset.

Dopo questo primo tracciamento, viene di nuovo rilevata la posizione del mouse, per stabilire se è stato mosso, e verso quale direzione, paragonando con una serie di *IF* le due rilevazioni di coordinate.

In base al risultato dei confronti, la nuova posizione viene aggiornata, ma solo di un pixel alla volta, in modo da ottenere una linea realmente continua.

Volendo accelerare il tracciamento sullo schermo, basterà spostare l'etichetta *muovi* due righe sopra al punto in cui si trova (come precisato nei commenti del listato), in maniera che il tracciamento avvenga esattamente in ogni nuova posizione, anche se distante dalla prima.

Così facendo, però, se per caso si muove il mouse con una certa velocità, la linea non risulterà più continua, ma frammentata (sperimentare per credere).

Unitamente alla determinazione della nuova posizione del mouse, e relativo tracciamento punto

per punto (o, se preferite, Line per Line), viene anche controllato il colore, o meglio la palette che lo identifica, "sopra" il quale ci si trova.

Allo scopo, il comando *Point* è veramente l'ideale.

Facile quindi, a seconda della palette riscontrata (= variabile P), provvedere ad eventuali penalità, tracciamenti o cambiamenti di colore dei riquadri neri.

A proposito di questi ultimi, c'è da dire che, ad ogni "contatto", è necessario paragonare l'attuale posizione del mouse (X ed Y), tramite un loop *For ... Next*, alle posizioni effettive di ogni numero nei riquadri, memorizzate nelle variabili con indice *X1()* ed *Y1()*.

Inizializzato il contatore (*count*), questo viene aggiornato se il quadrato è quello giusto, indicando ogni volta quale deve essere ristampato in arancione.

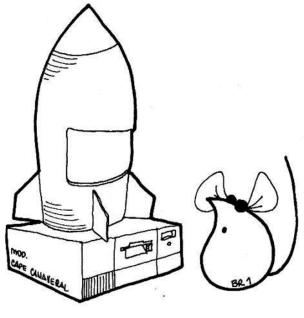
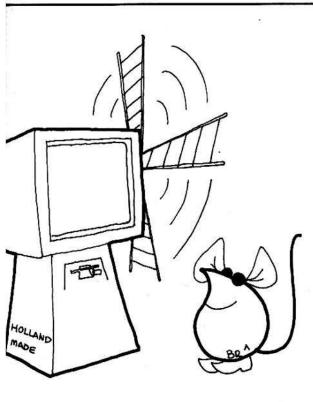
In effetti quest'ultima fase è più semplice (si fa per dire) da seguirsi direttamente sul listato, cercando di non perdere il filo nella lunga serie di If ed End If annidati uno dentro l'altro.

Tutto ciò che segue è semplice routine, non meritevole di ulteriori note.

Il programma, breve e conciso, non utilizza tecniche particolari ma, come si sarà notato, anche applicazioni apparentemente semplici possono portare ad approfondimenti più che degni di nota.

Se approfittarne o meno, è una decisione che spetta solo a voi.

Anche limitandosi al Basic, Amiga è in grado di dare notevli soddisfazioni

```
'========== INIZIALIZZAZIONI ==========

CLS:ON BREAK GOSUB chiude:BREAK ON
DIM flag(16),x1(16),y1(16),x2(16)
DIM y2(16),locx(16),locy(16)
start:
SCREEN 2,640,250,3,2:WINDOW 2,,,0,2
sp$=CHR$(32):count = 1:pu=0
x=5:y=12:a1$=SPACE$(5):a2$=SPACE$(2)
LOCATE 15,27:INPUT"DIFFICOLTA' (0-6)";tr
IF tr<0 OR tr >6 THEN start
PALETTE 4,1,,13,,93: CLS:COLOR 1,2

'===== DISEGNA SCHERMATA PRINCIPALE ====

FOR z1=0 TO 3
  FOR z2=1 TO 4
    caso:
    RANDOMIZE TIMER:xx=INT(RND*16+1)
    IF flag(xx)=1 THEN GOTO caso
    flag(xx)=1
'---- COORDINATE QUADRATI (IN PIXEL) ---
    x1(xx)=(y-1)*8:y1(xx)=(x-1)*8
    x2(xx)=x1(xx)+39:y2(xx)=y1(xx)+23
'--------------------------------------
    locx(xx)=x:locy(xx)=y
    GOSUB quadrati
    y=y+17
  NEXT z2
  y=12:x=x+6
NEXT z1
CIRCLE (310,120),30,4:PAINT(300,120),4,4

'=== RILEVAZIONE MOVIMENTO DEL MOUSE ===
'===== E TRACCIAMENTO DELLA LINEA =====

loop1:
IF MOUSE(0)=0 THEN GOTO loop1

'--------------------------------------
' Per una linea piu' veloce, ma meno
' precisa, inserire qui la label "muovi:"
'--------------------------------------

t=MOUSE(0):x=MOUSE(1)-8: y=MOUSE(2)-5
muovi:
LINE(x,y)-(x+tr,y+tr),1,bf
nolinea:
```

```
t=MOUSE(0):a=MOUSE(1)-8: b=MOUSE(2)-5
IF a>x THEN x=x+1:p=POINT(x+tr,y)
IF a<x THEN x=x-1:p=POINT(x,y)
IF b>y THEN y=y+1:p=POINT(x,y+tr)
IF b<y THEN y=y-1:p=POINT(x,y)

'===== CONTATTI TRA PALETTE DIVERSE ====

IF p=0 THEN GOTO muovi
IF p=1 THEN GOSUB contatto:GOTO muovi
IF p=3 THEN GOTO nolinea

'---- CONTROLLO SE QUADRATO VALIDO ----
```

```
IF p=2 THEN
 FOR xx=1 TO 16
  IF (x=>x1(xx) AND x<=x2(xx)) THEN
   IF (y=>y1(xx) AND y=<y2(xx)) THEN
    IF xx=count THEN
    COLOR 1,3:GOSUB quadrati:flag(xx)=0
    count=count+1:COLOR 1,2:GOTO nolinea
    END IF
   END IF
  END IF
 NEXT
GOSUB contatto:GOTO nolinea
END IF
```

```
'---- CONTROLLO SE BERSAGLIO VALIDO ----

IF p=4 AND count>16 THEN
  GOTO centro
  ELSE
  GOSUB contatto:GOTO nolinea
END IF

'----------------------------------------

GOTO muovi

'============  FINE GIOCO  =============

centro:
WINDOW 3,,(220,100)-(400,130),0,2
COLOR 0,4:CLS
LOCATE 2,5:PRINT "C E N T R O !!"
LOCATE 4,2:PRINT"ALTRO PERCORSO? (S/N)"
attesa:
a$=UCASE$(INKEY$)
IF a$="N" THEN GOTO chiude
IF a$="S" THEN
  sbr=1:GOSUB chiude
  RUN
END IF
GOTO attesa

'==========  SUBROUTINES  ==============

quadrati:
IF xx<10 THEN of$=sp$+STR$(xx)+a2$
IF xx=>10 THEN of$=sp$+STR$(xx)+sp$
LOCATE locx(xx),locy(xx):PRINT a1$
LOCATE locx(xx)+1,locy(xx):PRINT of$
LOCATE locx(xx)+2,locy(xx):PRINT a1$
RETURN

contatto:
LOCATE 1,1:pu=pu+1:PRINT "PENALITA' ="pu
RETURN

chiude:
WINDOW CLOSE 2
WINDOW CLOSE 3
SCREEN CLOSE 2
IF sbr=1 THEN RETURN
END
```

(a cura di **Domenico Pavone**)

MEA CULPA

□ Una sezione di posta solo per il miglior computer del mondo, il nostro Amiga, è stata una bella idea.

Però ho notato alcune "toppate", riferite al n.61 (pagina 17 e seguenti) della rivista, che meritano una correzione e una tirata d'orecchie all'autore (o chi per lui).

La prima riguarda la presunta inutilità dell'opzione *-debug* del comando *Loadwb*: il menu (nascosto) così abilitato è tutt'altro che inutile, anzi i suoi effetti sono decisamente interessanti. Effettuate infatti la seguente prova: attivate *CLI* (oppure *Shell*) e digitate:

loadwb -debug

Ora, senza uscire da ambiente CLI, clickate sull'icona di *Notepad* e caricate tutti i *Font* che potete; poi chiudete l'editor di testi e controllate la memoria libera rimasta. Ora accedete al menu a destra degli altri tre (senza intestazione visibile), selezionate *Flushlibs*, aspettate qualche secondo e ricontrollate la memoria libera: altro che inutile!

La seconda, è l'(im)possibilità del collegamento ad Amiga di Drive da 5.25 pollici. La cosa è possibile; nella *Mountlist* è presente l'*entry* per renderlo visibile al sistema, e sul disco *Extras* vi sono i programmi per usarlo in formato Ibm.

L'unico problema è che per cambiare disco bisogna prima cambiarlo fisicamente e poi eseguire il comando del Dos *DiskChange*.
(Flavio Stanchina - Trento)

• Riguardo al primo argomento, nulla da dire: hai ragione tu, anche se la mancanza di una visualizzazione del menu "-debug" nella *Title Bar* fa decisamente pensare ad una implementazione non ancora definitiva.

Con Flushlibs, infatti, come fai notare, viene allocato il maggior spazio di memoria possibile, rimuovendo il superfluo.

Selezionando invece *Debug*, a meno di non avere collegato un dispositivo seriale a 9600 baud (tipo stampante con ingresso RS232 o altro terminale), si provoca un inchiodamento della macchina.

A proposito dei Drive da 5 e 1/4, è vero che essi possono essere utilizzati, purchè debitamente "segnalati" al sistema attraverso il comando Mount (riferito ad una entry della Mountlist), come del resto qualunque altro tipo di periferica fisica o virtuale.

E' però anche vero, come confermato proprio dalla necessità di usare comandi come Mount o Diskchange, quanto affermato nelle pagine del n. 61, ovvero che la gestione di una simile periferica, non è "immediata" come quella dei drive da 3.5.

A parte l'obbligo di Diskchange, altri problemi possono anche insorgere nella compatibilità tra drive di diversa struttura: è per esempio impossibile usare Diskcopy se le due periferiche non sono dello stesso formato.

PERSONAL DISCHETTI

□ Attenendomi alle istruzioni dell'articolo "Cosa c'è sul disco Workbench" (n. 50), riesco a creare dischi personalizzati, ma con pochissimo spazio libero. Quali sono i files strettamente necessari al sistema?
(Francesco Baldacci - Livorno)

• Il vero problema non è sapere quali sono i files indispensabili al sistema, ma a quale uso è destinato il disco che si vuole personalizzare.

Per rendere autonomo un dischetto basta infatti formattarlo, installarlo, e quindi dotarlo delle seguenti directory e relativo contenuto:

S - startup-sequence
L - Disk-validator, Port-Handler, Ram-Handler.
LIBS - Icon.library, Info.library.
DEVS - System-configuration, parallel.device.
C - Run, Exexute, Loadwb.

Già così configurato, con una startup-sequence contenente il solo comando *Loadwb*, un dischetto è in grado di lanciare il sistema e creare l'ambiente Workbench, mantenendo tuttavia aperta la finestra Cli principale.

Volendola eliminare, ecco che già si rende necessario aggiungere qualcosa: il comando *Endcli*, da inserire nella directory C.

Se, poi, anche da Workbench si vuole mantenere la possibilità di formattare e copiare un dischetto, il sistema necessita anche di una directory *System*, contenente i comandi *Format* e *Diskcopy*.

Ma un professionista come Primo Giovedini non può permettersi nemmeno di trascurare l'eleganza delle sue linee in Basic...
Il basso punteggio ottenuto, solo 20 punti in totale, tiene conto anche di questo fatto...

E ancora, se si dispone di una stampante è indispensabile aggiungere nella directory *Devs* il file *Printer.device*, e creare (sempre all'interno di Devs) una subdirectory Printers con il file riservato al tipo di stampante adottata.

Gli esempi potrebbero continuare ancora a lungo, ma il concetto dovrebbe essere chiaro: se si dispone delle (minime) appropriate conoscenze, si può decidere di eliminare la maggior parte del contenuto originario del disco Workbench.

In caso contrario è più prudente non tralasciare troppi files, la cui mancanza potrebbe influenzare il corretto funzionamento di eventuali altri programmi o periferiche.

Un consiglio: ricavare le necessarie informazioni dai più svariati articoli, ma anche sperimentare di persona possibili soluzioni.

In fondo non è poi così complicato copiare il disco di sistema, e smanettarci dentro senza timore: nella peggiore delle ipotesi, la realizzazione di un'altra copia del Workbench non richiede poi grande fatica.

AMIGABASIC

☐ Avrei da porvi tre domande, tutte riguardanti AmigaBasic:

1) E' possibile "leggere" il colore contenuto in un punto dello schermo?

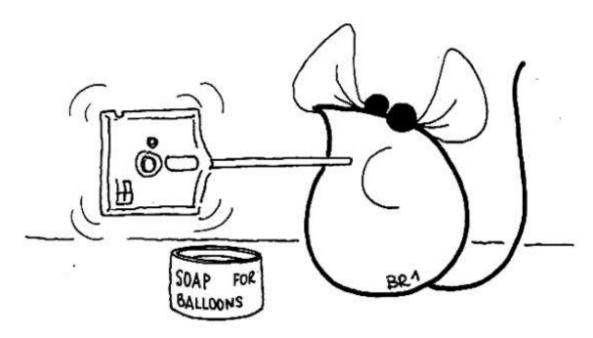
2) Esiste un comando per riportare i colori ai valori di default (come menu reset per i menu)?

3) Come si disattiva, in Basic, il filtro passa basso?

(Stefano Crimi - Caltanissetta)

• In stretto Amiga Basic, ovvero senza ricorrere a librerie di sistema, l'unico comando disponibile per leggere il contenuto-colore di un pixel, è *Point*.

In effetti si tratta di una lettura per così dire *indiretta*, in quanto non fornisce i reali parametri (intesi come percentuale di rosso, verde e blu) del colore presente in quel pixel, ma piuttosto il numero di Palette con il quale lo stesso è stato definito.

A titolo di esempio si digiti il breve listato, riportato in queste pagine, che, molto banalmente, disegna due figure piene, dopodichè precisa la palette che rappresenta i colori del pixel (all'interno di esse) indicato dalle linee rosse.

Provando a cambiare i parametri colore dei comandi *Line* e *Paint*, o posizionando *Point* altrove, si potrà sempre conoscere quale palette è attiva in quel preciso punto.

Per ripristinare, da programma, i valori di default delle Palette da 0 a 3 (sfondo, primo piano, ecc..), non c'è altro modo che ridefinirle una per una, riferendosi alla chiara tabella presente sul manuale in dotazione all'Amiga.

Quanto all'ultima (inflazionata) richiesta, l'Hi-fi sonoro si attiva con

POKE 12574721&, 254

...mentre si torna alle condizioni normali con...

POKE 12574721&,252

A seconda del modello Amiga (500 e 2000, o ROM differenti), oltre alla disattivazione del filtro audio, si avrà come conseguenza anche lo spegnimento, o l'abbassamento di intensità, della spia Power.

FUSIONI CALDE

☐ Come si possono fondere due routine in ABasic?

(anonimo... non veneziano)

• Anzitutto, a scanso di equivoci, va precisato che Amiga Basic (dal resto della missiva sembra evi-

dente che il lettore si riferisce ad esso) non può essere abbreviato in *Abasic*, in quanto esiste un altro interprete proprio con questo nome, prodotto dalla Metacomco: la Microsoft potrebbe... soffrirne.

Quanto al *merge* di due routine, è possibile realizzarlo tramite il comando Merge (guarda caso!), specificando come parametro (tra virgolette) il nome del file da aggiungere a quello già presente in memoria.

Unica accortezza, il file da "appendere" deve essere in formato Ascii, ovvero precedentemente salvato digitando in modo diretto (e *non* dall'apposita opzione nel menu project):
Save "nomefile", A

```
SCREEN 2,320,250,3,1
WINDOW 3,,,0,2
PALETTE 0,0,0,0
PALETTE 1,.93,.2,0
PALETTE 2,.33,.87,0
PALETTE 3,1,.13,.93
PALETTE 4,.4,.6,1
PALETTE 5,1,.6,.67
LINE (35,35)-(170,170),5,bf
CIRCLE (100,100),50,4
PAINT (100,100),3,4
'-----------------
x=POINT (100,100)
y=POINT (150,60)
'-----------------
LINE (100,100)-(180,100)
LINE (150,60)-(180,60)
LOCATE 8,24:PRINT"palette"y
LOCATE 13,24:PRINT"palette"x
LOCATE 25,7:PRINT"PREMI UN TASTO"
attesa:
a$=INKEY$
IF a$="" THEN GOTO attesa
WINDOW CLOSE 3
SCREEN CLOSE 2
```

EMULATOR 64

□ **Adoperando il programma "64 Emulator" senza collegarlo ad un drive 1541 (che non posseggo), ho notato che dalla modalità C/64 non si riesce ad accedere al drive di Amiga nè alla Ram, quale che sia il numero di device loro assegnato.**

Anzi, si provoca un inchiodamento della macchina.

E' un problema legato alla non originalità della mia copia del programma o altro?
(Patrizio Colaruotolo - Gaeta)

• Il programma in questione, in effetti, può presentare diversi problemi, legati tanto alla (quasi) impossibilità di trovare una copia originale, quanto alla introduzione delle nuove Rom 1.3.

Comunque, secondo un parere strettamente personale, l'unico motivo che possa giustificare l'utilizzo degli emulatori 64 in circolazione, è la possibilità di trasferire dei files (soprattutto di tipo Ascii) dai drive 1541/71 all'Amiga.

Cosa che, ovviamente, può interessare chi ha molto "sudato" sul buon vecchio 64 (o 128).

Tra l'altro, il trasferimento è reso possibile dal programma Transfer, (spesso) presente nello stesso disco dell'emulatore, ma totalmente indipendente da esso.

A parte questa fetta di utenti per così dire "reduci", e con motivazioni pratiche, è proprio da masochisti (o da sadici, nei confronti della macchina) ingabbiare un Amiga nei ben miseri confini di un Basic 2.0, o dei nostalgici "38911 bytes free".

UN SISTEMA A PROVA DI GURU

Con le nuove possibilità offerte dal Dos 1.3, è finalmente possibile rendere ultraveloce l'inizio dei lavori, affrancando il nostro Amiga dalla noiosa schiavitù del boot da disco

di **Gregor Samsa**

Un batch file è sufficiente a rendere "immortale" la boot-sequence di Amiga

C'era una volta VdO...
Così potrebbe iniziare una novella neo-amighevole, debitamente provvista di finale a lieto fine: la diffusione ufficiale del Workbench 1.3, e relative Rom per i modelli 500 e 2000.

Per chi non lo sapesse, VdO, *Virtual Disk*, una realizzazione pressochè di pubblico dominio (l'autore, ignaro dell'italico costume, richiedeva ingenuamente 10 dollari attraverso la schermata di presentazione), è stato uno dei primi veri comfort per i più assidui frequentatori di *Workbench* e *Cli*.

Consentiva, infatti, di sfruttare una *Ram Disk* che non veniva cancellata dall'operazione di reset, ed il suo contenuto poteva quindi essere recuperato anche dopo il più "tosto" dei Guru.

Sin dalla pre-release del disco Workbench 1.3, questa possibilità si è finalmente allargata a tutta l'utenza di Amiga, anche se dotata delle Rom 1.2, con l'introduzione della *Rad*, una *Recoverable Ram* dalle notevoli prestazioni.

Come si può intuire, i vantaggi derivanti da una simile implementazione sono enormi.

E' noto, infatti, l'uso massiccio che spesso si fa della Ram Disk per velocizzare molte operazioni dell'Amiga Dos, o semplicemente per immagazzinarvi qualche programma di uso più frequente, il cui accesso diventa, in tal modo, praticamente immediato.

Altrettanto noto è il fatto che la provvisorietà del device *Ram:* spesso scoraggia la memorizzazione di dati importanti, o l'assegnazione, a questa (virtuale) periferica, di directory piuttosto lunghe da copiare (tipo la C): non è decisamente pratico rifare tutto da capo se interviene una dannata Guru Meditation o si ha la necessità di resettare il sistema.

Con una Recoverable Ram, il problema viene totalmente eliminato, anche disponendo delle Rom 1.2.

Chi, invece, ha il sistema aggiornato alla versione 1.3, può accedere alla più potente caratteristica di questa nuova struttura: la capacità di autoboot.

Ma non precorriamo i tempi, e diamo un'occhiata ravvicinata alla Ram Disk in grado di "resuscitare".

SE CI SEI, BATTI UN COLPO

Mettiamo, dunque, in moto il nostro mostriciattolo a 16 bit, e, indipendentemente dalla versione del Kikstart, attiviamo il sistema con il disco Workbench 1.3.

Quando, a conclusione della lunga serie di ronzii del drive, sarà (finalmente) apparso il consueto schermo di lavoro, potremo constatare la presenza delle solite icone, ovvero quella del dischetto inserito nel drive (qualora non se ne possegga anche un secondo), e quella della nota Ram Disk.

Di Recoverable Ram, neanche l'ombra.

Il fatto è che ad installarla dobbiamo pensarci noi, in quanto nella startup-sequence del disco di sistema non ne è prevista l'implementazione.

Diamoci dunque da fare: apriamo l'icona-disco, assestiamo due bei "click" sulla Shell (se avete il Cli, fa lo stesso), e spostiamo la nuova finestra in modo da lasciare libera alla vista la barra dello schermo

```
;--------------------------------------.
:              F I L E    1
;--------------------------------------
resident cli l:shell-seg system pure add
resident c:execute pure add
mount newcon:
list >nil: ram:
setmap i
FF
newshell newcon:0/95/640/140/RAD.SYSTEM
loadwb -debug
endcli
```

Prima di trasferire nella Ram disk i vari file, rintracciate con precisione le directory che li contengono

Workbench e la porzione destra dello stesso (dove sono le icone, per intenderci).

La normale Ram Disk, come è noto, basta citarla con qualsiasi comando perchè essa venga automaticamente installata e, se è attivo il Workbench, anche iconizzata (nel nostro caso, ci ha già pensato la startup-sequence originale del disco sistema).

La Rad, invece, occorre prima "montarla".

Per avere le idee più chiare, cominciamo intanto a digitare, nella finestra Shell, *Info* (+ Return).

Verrà così mostrata l'attuale condizione delle periferiche al momento installate.

Ora impartiamo *Mount Rad:* (non dimenticate il doppio punto finale!): a parte il ronzio del disco, non succederà assolutamente nulla, come verificabile con un altro comando Info.

Dopo Mount, infatti, a meno di non agire sulla *Mountlist* (la vedremo tra breve), è necessario procedere come per la Ram Disk, ovvero impartire un qualsiasi comando che faccia riferimento alla Rad.

Quindi, per esempio, digitiamo *List Rad:* ed osserviamone l'effetto.

Un messaggio del Dos comunicherà che la directory richiesta è *empty* (cioè vuota: grazie, ma lo sapevamo) e, quasi immediatamente, sullo schermo Workbench apparirà una nuova icona: *RamBO* (liberamente rinominabile).

Era ora!

In (s)compenso, come si può constatare clickando fuori dalla finestra Shell, la memoria libera avrà subito un massiccio decurtamento.

Un ulteriore comando Info confermerà l'esistenza del nuovo device, la cui dimensione risulterà essere di ben 242 KRam.

Ecco, quindi, una prima palpabile differenza con la tradizionale Ram Disk.

Quest'ultima è *elastica*, nel senso che aumenta le sue dimensioni a seconda di ciò che si pone dentro: risulta, infatti, sempre "piena" quando si apre la sua icona (vedi barra a sinistra della sua finestra).

La Rad, di contro, ha una struttura per così dire *rigida*, come un vero e proprio disco, e può quindi (in questo caso) memorizzare 242 KRam e non di più: aprendone infatti l'icona, la si vedrà vuota fintanto che non si comincerà a sfruttarla.

Se si desidera liberare la memoria dalla presenza della Rad, basta impartire *RemRad*, badando che, alla scomparsa della sua icona (e relativo incremento della memoria "free"), non corrisponde una totale eliminazione, che avverrà solo dopo un reset.

La cosa, comunque, assume importanza solo se si tenta di "rimontare" la Recoverable Ram dopo averla estirpata con RemRad: se non si è provveduto al reset, il sistema segnalerà un "Rad: already mounted" (misteri di progettazione!).

Inutile aggiungere che, da un punto di vista pratico, la device Rad: va usata esattamente come la normale Ram Disk, tanto da Cli che da Workbench.

Giusto per verificarne l'efficacia, si può provare (da WB) ad aprirne l'icona e trasferirvi qualcosa (per esempio l'orologio o Notepad) dal disco di sistema.

Si dia poi un Reset (Control, Amiga, *Amiga*), e, dopo avere rimontato la Rad, si controlli il suo contenuto con List o aprendo la sua icona: l'orologio deve trovarsi ancora lì.

Per non dover ripetere dopo ogni reset la scomoda sequela di operazioni, il comando Mount può inoltre essere inserito nella startup-sequence, ma anche di questo se ne riparlerà più avanti.

TUTTO COMPRESO

Quanto finora visto è valido per qualunque release o configurazione di memoria di Amiga (con esclusione di eventuali 1000 non espansi), ma una prima discriminante è già evidente: con soli 512K di base, una volta installata una Rad di 242K, rimane troppo poco spazio per altre applicazioni degne di nota.

Di contro, chi ha la fortuna (quasi obbligata) di godersi almeno un Megabyte, può trovare utile ri-

servarsi una fetta più consistente di Ram "indelebile".

Ebbene, in entrambi i casi qualcosa si può fare.

L'installazione della Recoverable Ram, infatti, è legata a tre elementi software, indispensabili se (ad esempio) si vuole dotare un altro dischetto della possibilità di sfruttarla:

1) Il comando *Mount*, presente nella directory C.
2) Il file *Ramdrive.device*, inserito nella directory *Devs*.
3) Il file *Mountlist*, rintracciabile anch'esso in Devs.

Il terzo, in particolare, può essere modificato ad uso e consumo dell'utente, in funzione di eventuali adattamenti al proprio sistema.

Diamogli subito un'occhiata, ma, per precauzione, si faccia prima una copia del disco Workbench: un intervento scorretto potrebbe danneggiare l'originale.

Utilizzando il disco appena copiato, apriamo dunque una finestra Shell o Cli, e digitiamo...

Cd Sys:Devs

...in modo da "trovarci" nella directory voluta.

Per leggere la Mountlist, ed eventualmente modificarla, sfruttiamo uno degli editor inclusi nel disco-sistema.

Più in concreto, si digiti...

Ed Mountlist

Ora si scorra il file agendo sui tasti cursore, finchè non apparirà una riga che inizia con *RAD:*

Eccoci a destinazione.

Tutto ciò che segue (fino al simbolo "#33), definisce i parametri del device Rad.

Cominciamo col rendere meno complicata la procedura di "montaggio".

Ci si porti sull'ultima riga, quella contrassegnata dal cancelletto, si prema Return per far spazio ad una nuova linea, e si digiti *Mount = 1* allineandola, se lo desiderate, con le altre precedenti.

Con tale aggiunta, quando in seguito si impartirà il comando *Mount Rad:*, questa verrà immediatamente installata, senza la necessità di "citare" prima la periferica, come visto in precedenza.

Le dimensioni del disco virtuale sono precisate

```
;-----------------------------------------
;                F I L'E   3
;-----------------------------------------
path system add
resident CLI l:shell-seg System pure add
resident c:execute pure add
mount newcon:
ff
setmap i
ask "installo sistema in RAD? (y/n)"
if not warn
 skip cont
endif
execute sysrad
quit
lab cont
copy disk.info to ram:
loadwb -debug
endcli
```

dalla entry *HighCyl = 21* che, come se si trattasse di un vero e proprio disco, stabilisce che la periferica Rad: dispone di 21 tracce.

Modificando tale valore si modifica anche l'occupazione di memoria della Recoverable Ram.

Si possono tentare vari esperimenti tenendo presente che, ad esempio, con HighCyl = 15 si otterrà una Rad di 176 K mentre, usando 40 tracce, l'occupazione si incrementerà fino a 451 K.

Apportate le modifiche volute, non resta che uscire da Ed (con *Escape + X + Return*), che riscriverà la nuova Mountlist sul dischetto.

Per gli "Amighi" in versione 1.2, la cosa finisce qui.

Per i più aggiornati (release 1.3), invece, il meglio deve ancora venire.

Una memoria indelebile evita perdite di tempo inutili

UN PO' DI PRATICA

Vi consigliamo di leggere interamente il presente paragrafo *prima* di passare alla digitazione dei listati descritti.

Cominciamo subito (si veda *anche* il riquadro) con il creare un po' di spazio nel disco Workbench, ovvero nella sua *copia*, eliminando il cassetto Utilities. Forse non riuscirete ad eliminarlo completamente, ma su questo argomento torneremo un'altra volta.

Ora, utilizzando *Ed*, si copi il File 1, pubblicato in queste pagine, che rappresenterà la futura startup-sequence della Rad, e lo si memorizzi, con nome *RadStart*, nella directory S (= digitare *Cd S* prima di cominciare) del disco-sistema.

Banalmente, questo batch file consente di aprire una finestra Shell, nonchè uno schermo Workbench. Inutile dire che il nome impostato dovrà essere rigorosamente eguale a quello citato (*RadStart*) perchè lo ritroveremo nel secondo file da digitare (vedi File 2, più o meno nella parte centrale).

Inoltre, ma solo a titolo di esempio, con *Setmap i* si rende operativa la tastiera italiana.

Chi non necessita di tale accorgimento non deve copiare il suddetto comando (nè, in seguito, quello presente, *setmap*, verso la fine del secondo file).

Precisiamo che tale file può trovarsi, a seconda del WB posseduto, non solo in directory diverse da quelle menzionate, ma addirittura nel disco *Extras*; al lettore l'arduo compito di rintracciare il file in oggetto e di effettuare le dovute modifiche.

Ultimata questa fase tocca ora al File 2, leggermente più lungo.

Questo, infatti, si preoccupa di copiare (vedi anche C.C.C. n. 63) tutto il necessario dal disco Workbench alla Rad, compreso il File 1 precedentemente creato, che assumerà ora il suo vero nome: *startup-sequence*.

Anche questo file dovrà essere memorizzato nella directory S, in modo da essere poi attivato tramite un semplice *Execute* (comando indispensabile per attivare i batch file).

Come nome, gli si può assegnare *Sysrad*.

All'interno del file alcune sue parti possono (o devono) essere modificate.

E' il caso della directory *Keymaps* e del file *"i"*, eliminabili se il vostro Amiga possiede la tastiera americana (e, ripetiamo, se i suddetti file non sono presenti nel disco copia di WB).

Inoltre viene copiato, nella directory devs/printers, il file *generic*, riferito alla stampante adottata (se non possedete la stampante, potete evitarne la digitazione).

Se con *preferences* si è selezionato un altro modello, è necessario specificarne il nome al posto di Generic.

Della directory C vengono qui copiati solo i comandi strettamente indispensabili alla startup-sequence, ma in una Rad di 242 K rimane abbastanza spazio per memorizzarne parecchi altri: a voi la scelta, inserendoli eventualmente in coda al File 2, come mostrato dalle righe libere precedute dal punto e virgola (simbolo di *commento*).

Chi dispone di più memoria, inoltre, può inserire anche una directory System con i comandi *Format* e *Diskcopy*, sempre che si preveda di lavorare maggiormente in ambiente Workbench: in caso contrario, è sufficiente aggiungere i soli comandi nella solita C.

Conclusa anche questa fatica, basta ora impartire (sempre da Shell o Cli) il comando...

Execute Sysrad

...per avere pronta all'uso la Recoverable Ram.

E' bene notare che se, in caso di errore, la procedura Execute dovesse interrompersi per qualche motivo (errata digitazione dei listati, file da copiare non presenti sul vostro WB oppure presenti in altri "cassetti" del WB di cui si è effettuata la copia e così via), sarete forse costretti a spegnere il computer, attendere un adeguato intervallo di tempo ed accenderlo nuovamente ricominciando tutto daccapo. In caso contrario, infatti, può capitare che la Recoverable Ram risulti già installata e la procedura Execute (che ne prevede l'installazione con *mount rad:*) verrebbe costantemente interrotta!

Si può inserire il comando *Execute Sysrad* nella startup-sequence del disco Workbench, magari con una possibilità di scelta sull'installarla oppure no, come viene mostrato nell'esempio proposto dal File 3.

Si tenga presente, in questo caso, che il batch file richiederà, durante la sua esecuzione, un accesso in scrittura al disco, da non interpretare come possibile attacco... batteriologico.

In definitiva, avremo: il file 3 come startup-sequence del disco di sistema, il file 1 (a nome Radstart) ed il file 2 (a nome Sysrad) nella directory S dello stesso disco.

Se tutto è in regola, non resta che attivare il file 2 direttamente (con Execute Sysrad) o, se si è scelto di inserirne l'esecuzione nella procedura di avvio, rispondendo "Y" alla richiesta.

In entrambi i casi, dopo un'attesa che verrà ampiamente ripagata, alla comparsa della segnalazione "Rad pronta!", sarà sufficiente sfilare il disco dal drive, e resettare il sistema.

Pochi secondi, ed ecco un nuovo Workbench ed una finestra Shell tutti per noi, pronti ad obbedire a qualsiasi comando, e, soprattutto, ripristinabili in

```
; ---- - -.-- - ---------------
;        F I L E   2
;----------------------------
CD sys:
mount rad:
copy c/copy ram:copy
copy c/assign ram:assign
assign y: ram:assign
Y: x: ram:copy
makedir rad:L
y: z: rad:L
x: L/disk-validator z:
x: L/Port-handler z:
x: L/Shell-seg z:
x: L/Newcon-handler z:
x: L/Ram-handler z:
makedir rad:devs
y: z: rad:devs
x: devs/mountlist z:
x: devs/printer.device z:
x: devs/parallel.device z:
x: devs/ramdrive.device z:
x: devs/system-configuration z:
makedir z:printers
makedir z:keymaps
x: devs/keymaps/i z:keymaps
x: devs/printers/generic z:printers
makedir rad:s
x: s/radstart rad:s/startup-sequence
makedir rad:libs
y: z: rad:libs
x: libs/icon.library z:
x: libs/info.library z:
makedir rad:c
y: z: rad:c
cd C
x: endcli z:
x: execute z:
x: ff z:
x: loadwb z:
x: mount z:
x: newcli z:
x: newshell z:
x: resident z:
x: run z:
x: setmap z:
x: list z:

;x:.............
;x:.............
echo "RAD PRONTA! Resetta il computer"
echo "dopo aver tolto i dischetti"
```

qualunque momento con la perdita di un'altra (misera) manciata di secondi.

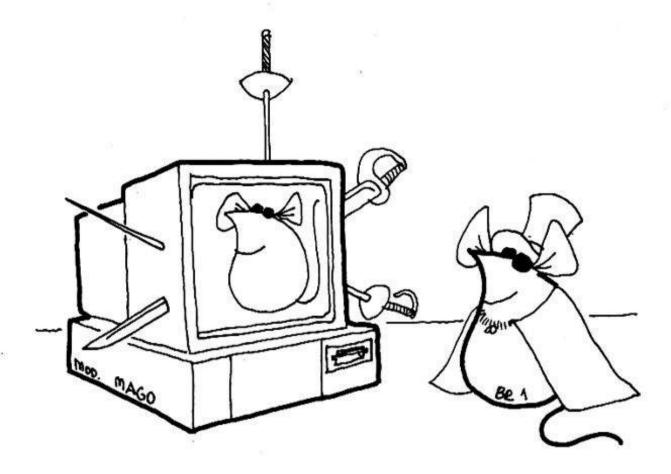
Il tutto senza l'ausilio di alcun dischetto, che anzi non bisogna dimenticare di togliere dal drive: in caso contrario, sarebbe lui a prendere il sopravvento nella procedura di boot.

Teniamo a sottolineare che alcuni comandi tipici del Dos (tra cui, soprattutto, CD e DIR) non sono stati inseriti nella sequenza finale del file 2. Ciò allo scopo di spingervi ad esaminarlo con la *massima* attenzione prima di digitarlo e mandarlo in esecuzione. Da notare, e lo ripetiamo ancora una volta, che alcuni file potrebbero esser presenti in cassetti diversi da quelli da noi indicati; ad esempio *setmap*, nel nostro caso, lo supponiamo presente presente nella directory C. Nel vostro dischetto potrebbe trovarsi nel cassetto "Systems" o, addirittura, in una delle directory del disco Extras. Vi conviene, pertanto, esaminare con cura il listato pubblicato per verificare che, in fase di esecuzione, non abbiate amare sorprese.

Da ricordare, infine, che *prima* di impartire il comando *endcli* è necessario digiatare *loadwb* per consentire la visualizzazione delle icone dei dischetti. In caso contrario, infatti, non potreste proseguire nel lavoro!

Prima di concludere, si ricordi che nel Dos 1.3 esiste il comando *IconX*, sufficientemente descritto nel manuale in dotazione ad Amiga: potrebbe essere una buona idea, nonchè un ottimo esercizio, dotare di icona il file 2 (senza più memorizzarlo in S), rendendolo attivabile direttamente da Workbench.

Bene, sulla Rad non si è forse detto proprio tutto, ma ce n'è abbastanza per cominciare a trattarla più confidenzialmente.

E, magari, darle del tu.

UNA RAM DISK DAVVERO SUPER

Il modo migliore per verificare le nuove capacità della Recoverable Ram, se implementate in congiunzione con le Rom 1.3, consiste in un breve esperimento.

Dopo aver installato la Rad secondo le modalità descritte nell'articolo, si provi a togliere il dischetto dal drive 0, e quindi si resetti il computer da tastiera, senza spegnerlo.

Dopo qualche secondo apparirà, al posto della solita immagine con dischetto e... pollicione V1.3, una bella finestra Cli.

Niente paura, non si tratta di qualche nuovo virus sconosciuto, anche se qualunque comando digitato nella finestra non sortirà alcun effetto.

Il fatto è che il sistema, dopo aver cercato (inutilmente) di trovare qualcosa in *df0:*, se è presente una device Rad, provvede ad attivare la procedura di boot da quest'ultima.

Nel nostro caso, non trovando la dovuta startup-sequence all'interno di essa, abortisce la procedura, rimanendo con la finestra Dos aperta.

Le potenzialità di quanto visto, sono davvero enormi, e di una comodità che finora sembrava appartenere solo alla fascia inferiore dei computer ad otto bit.

Basterà, infatti, far trovare ad Amiga un sistema (più o meno) completo in Rad:, perchè esso diventi punto di riferimento per tutte le varie assegnazioni, Sys: compresa.

Si otterranno, in pratica, due effetti: anzitutto, i disk drive rimarranno completamente liberi per l'utente, senza più i continui requester che impongono l'inserimento del disco di sistema.

Inoltre, in caso di eventuali Guru o Reset, basterà disinserire i dischetti presenti nel drive, per avere in pochi secondi di nuovo il sistema attivo, Workbench o Cli che sia, con una velocità assolutamente non paragonabile a quella del boot da disco.

In sintesi, per ottenere queste prestazioni, tutto ciò che dobbiamo fare è:

1) Fornire la Rad: di una startup-sequence, ovviamente contenuta in una directory S in modo da garantire l'automatismo dell'avviamento.
2) Inserire le directory L, Libs e Devs con i files indispensabili al sistema.
3) Aggiungere infine la directory C, con i comandi contenuti nella startup-sequence, più quelli eventualmente necessari (Run ed Execute, sempre!).

Ovviamente, per un simile utilizzo, le dimensioni della Rad devono essere adeguate, il che, tradotto in parole povere, significa che non si può andare molto al di sotto di 242K.

UN MESSAGGIO A CARATTERI CUBITALI

Chi ha la stampante potrà sfruttare la rotine l.m. (interamente rilocabile!) per riportare su carta messaggi di ogni forma e dimensione

di **Tito Brasolin**

Fra le caratteristiche principali delle stampanti di elevata qualità vi è la possibilità di stampare caratteri di dimensioni raddoppiate o quadruplicate, anche in altezza. Programmi come Geowrite, poi, consentono di selezionare font diversi di caratteri.

Il programma in linguaggio macchina proposto consente di realizzare qualcosa di simile (e di meglio): definita una stringa (B$), viene mandata in stampa variando, eventualmente, le dimensioni dei caratteri (con un minor numero di limitazioni sia rispetto alle stampanti in commercio sia al programma "professionale" citato). Il loro "aspetto" è tratto direttamente dalla memoria carattere del computer e, di conseguenza, anche da altri set di caratteri ridefiniti, eventualmente presenti in Ram al momento dell'esecuzione.

Un carattere stampato avrà dimensioni minime di 8 x 8 dot (puntini elementari). In orizzontale, pertanto, un comune foglio di carta non ne conterrà più 80 (come siete abituati con un normale List) ma, AL MASSIMO, 60. Se avete bisogno dei "vecchi" caratteri non dovete fare altro che ricorrere al familiare comando Print#.

La routine funziona con qualsiasi stampante MPS-803 compatibile.

La sintassi di attivazione della routine è la seguente:

SYS xxxxx, "testo", fx, fy, lm, xl [,carset [,device]]

...in cui:

XXXXX: indirizzo di partenza qualsiasi: ricordiamo che la routine è completamente rilocabile (attenzione, comunque, al Top di memoria!).

"TESTO": stringa da mandare in stampa. Può essere espressa esplicitamente o contenuta in una variabile stringa. Se è nulla, ovviamente, non sarà stampato niente. I caratteri di controllo che su schermo appaiono in negativo verranno stampati in positivo.

FX: fattore di ingrandimento orizzontale. I nuovi caratteri avranno una larghezza in pixel pari a 8*FX. Tale parametro deve essere compreso tra 1 e 60.

FY: fattore di ingrandimento verticale. I nuovi caratteri avranno altezza pari a 8*FY. Tale parametro è compreso fra zero e 255, con lo zero che vale come 256.
LM: margine sinistro. E' un numero compreso fra 0 e 59 (la nostra stampante, infatti, diventa un modello a sessanta colonne...).

XL: larghezza delle righe. Il testo da mandare in stampa viene diviso in sezioni di lunghezza xl; al termine di ciascuna di queste la stampante va a capo automaticamente. XL non può essere superiore a int ((60 - lm) / fx) ma, qualora lo fosse, verrebbe automaticamente posto uguale a quest'ultimo valore.

Questi primi parametri sono indispensabili per il funzionamento della routine.

E' ancora da precisare, al loro riguardo, che se fx fosse tale da non consentire la stampa neanche di

La routine è un po' lunga, ma vale la pena digitarla

```
100 REM * STAMPA DI MESSAGGI GIGANTI
110 REM * BY TITO BRASOLIN
120 REM * GIUGNO 1989
130 REM * DIMOSTRAZIONE (C 64 + MPS 803 COMPATIBILE)
140 REM * ALLOCARE ROUTINE DA 49152 !!!
150 :
160 POKE 650,128 :REM RIPETIZIONE TASTI
170 PRINT CHR$(147):"USA I TASTI FUNZIONE PER MODIFICARE I   PARAMETRI"
180 PRINT:PRINT "F1-F2 PER FX      F3-F4 PER FY"
190 PRINT "F5-F6 PER LM      F7 PER STAMPARE"
200 PRINT:INPUT "BYTE ALTO SET CARATTERI  208█████";CS
210 INPUT "DEVICE  4███";DV
220 INPUT "CONTROLLO PER MAX XL (S/N)  S███";C$
230 FX=1:FY=1:PRINT
235 B$="VIVA IL C 64":INPUT"STRINGA";B$
240 PRINT CHR$(147);
250 XL=INT((80-LM)/FX)
260 IF B$<>"" AND C$="S" AND LEN(B$)>XL THEN B$=LEFT$(B$,XL)
270 PRINT "FX:"FX,"FY:"FY,"LM:"LM,"MAXXL:"XL
280 PRINT CHR$(18);B$
290 GETA$:IFA$=""THEN290
300 IF A$=CHR$(133) THEN FX=FX+1
310 IF A$=CHR$(137) THEN FX=FX-1
320 IF A$=CHR$(134) THEN FY=FY+1
330 IF A$=CHR$(138) THEN FY=FY-1
340 IF A$=CHR$(135) THEN LM=LM+1
350 IF A$=CHR$(139) THEN LM=LM-1
360 IF A$=CHR$(20)ANDB$<>""THENB$=LEFT$(B$,LEN(B$)-1)
370 IF A$=CHR$(136) THEN 470
380 IF FX>80 THEN FX=1
390 IF FX<1 THEN FX=80
400 IF FY>255 THEN FY=1
410 IF FY<1 THEN FY=255
420 IF LM>59 THEN LM=0
430 IF LM<0 THEN LM=59
440 IF ASC(A$)<32 OR ASC(A$)>127 THEN240
450 IF LEN(B$)<255 THEN B$=B$+A$
460 GOTO 240
470 SYS 49152,B$,FX,FY,LM,XL
480 GOTO 240
```

un carattere a partire da Im, il programma terminerebbe senza stampare nulla.

Inoltre non è possibile fare coesistere, su una stessa riga, caratteri di dimensioni diverse, oppure caratteri in positivo e in negativo o maiuscoli e minuscoli contemporaneamente.

I codici Ascii della stringa, per consentire i necessari calcoli, vengono convertiti in codici schermo compresi tra 0 e 127 in accordo con la tabella riportata in queste pagine.

Per variare il modo di stampa è necessario agire sul parametro (facoltativo) "Carset".

CARSET: byte alto della locazione iniziale del set di caratteri prescelto. Per default il suo valore è 208 (208 * 256 = 53248, inizio memoria carattere) consentendo la stampa in modo maiuscolo + grafico / positivo.

Altri valori utili sono:
212 = modo maiuscolo + grafici / negativo
216 = modo minuscolo + maiuscolo / positivo
220 = modo minuscolo + maiuscolo / negativo

L'utilizzo più interessante di Carset è comunque nella stampa di caratteri ridefiniti dall'utente; è sufficiente indicare il byte alto della locazione d'inizio del set riprogrammato (a patto che i dati non si trovino sotto la ROM del BASIC, mentre possono trovarsi sotto il KERNAL).

E' evidente che l'inizio dei dati relativi a un carattere di un set si trova con:
(codice schermo) * 8 + carset

L'ultimo parametro (facoltativo) è il device number, posto per default a 4.

LA TECNICA USATA

Un problema sorto nel realizzare il programma consisteva nella necessità di far "passare" la testina di stampa più volte su righi successivi, in modo da completare l'ingrandimento verticale. L'uso del modo di stampa "bit a bit" (che può stampare al massimo 7 bit in verticale, mentre un carattere può essere alto otto, o più, dot) è stato quindi il principale ostacolo.

La soluzione scelta è risultata un po' macchinosa, ma efficace. Nel registro X è stato inserito un contatore alla rovescia per il fattore di ingrandimento verticale; in Y un indice che punta al byte in corso della matrice del carattere da ingrandire. Si carica, poi, un byte di base col valore binario 01000000.

Con una maschera appropriata viene testata una colonna della matrice del carattere, e vengono fatti ruotare (nel byte di base, da sinistra a destra) tanti bit (accesi o spenti a seconda del risultato del test) quanti ne indica il registro X.

L'operazione ha termine quando il bit inizialmente acceso del byte di base "esce" fuori (cioè il carry si accende). Dopo un'ultima rotazione verso destra si ha a disposizione un byte pronto per essere mandato in stampa tante volte quante ne indica il fattore di ingrandimento orizzontale.

La stessa procedura viene compiuta per tutte le "colonne" del carattere e per tutti i caratteri della riga, sempre partendo dagli stessi valori iniziali dei registri X e Y.

Se la riga risulta completata, si va a capo e si riparte dal primo carattere in modo analogo a quanto già svolto; ora, però, i nuovi valori iniziali di X e Y sono quelli con cui è stata conclusa la "passata" precedente.

Il termine (anche in verticale) di una riga è segnalata dall'indice Y, che contiene un valore superiore a 7. Se ciò avviene, gli ultimi bit del byte da mandare alla stampante vengono comunque posti a zero. Il numero di questi ultimi bit nulli è variabile, tuttavia non è mai superiore a 6 e spesso genera uno "stacco", rispetto alla riga successiva, che contribuisce ad aumentare la leggibilità.

Se interessa conoscere quanti sono esattamente i bit "morti", calcolate 8*fy e verificate quante unità separano tale valore dal primo multiplo di sette, maggiore o uguale ad esso.

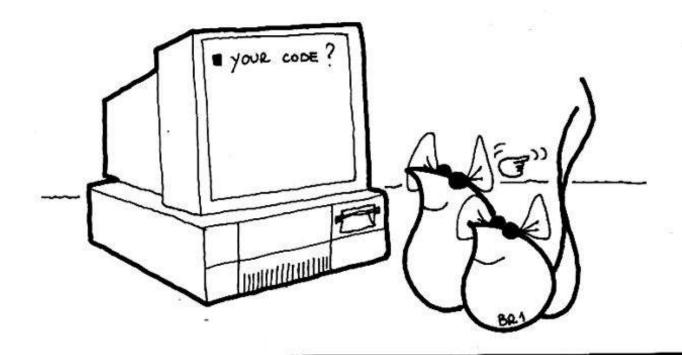
Per esempio: fy=2 , 8*2=16 , 21-16=5 bytes morti. Oppure fy=7 , 8*7=56 , 56-56=0 bytes morti.

CONTATORI E PUNTATORI

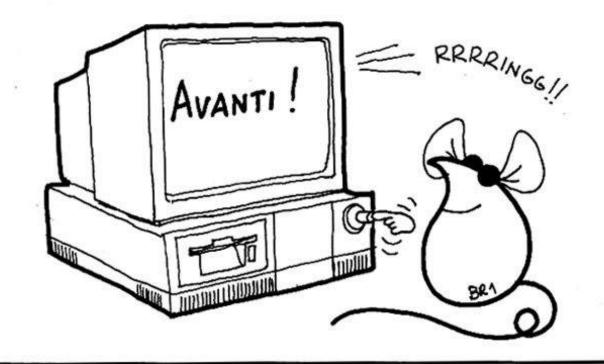
Un ultimo cenno su alcuni contatori e puntatori usati:

CNTLIN = indice che punta al carattere in corso. Viene aggiornato ad ogni carattere. Il suo valore iniziale è CNTLEN.

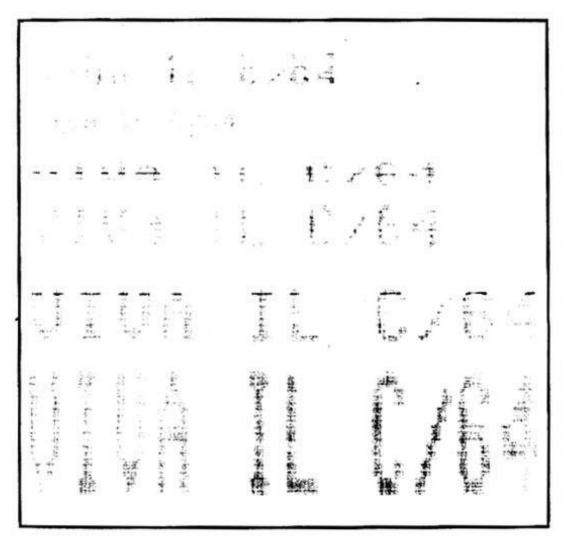
CNTLEN = indice che punta al primo carattere della riga in corso di stampa. Viene aggiornato solo quando la riga è completa anche in senso verticale.

La stampa risulta velocissima grazie al linguaggio macchina

```
100 REM              "CARATTERI CUBITALI"
110 REM PROGRAMMA PER C 64 + MPS-803 (E COMPATIBILI)
120 REM IN LINGUAGGIO MACCHINA INTERAMENTE RILOCABILE
130 REM
150 REM PER UTILIZZARLO, LEGGI LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
180 Y=-1:GOSUB 1000:PRINT:X=49152:REM DEFAULT
185 INPUT"DA QUALE LOCAZIONE DEVO ALLOCARE";X
190 READW:Y=Y+1:IFW<0THEN210
200 GOTO190
210 PRINT"PRIMA LOCAZIONE ="X
220 PRINT"ULTIMA LOCAZIONE ="X+Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GETA$:IFA$=""THEN240
250 IFA$="S"THENRESTORE:T=X:GOTO290
260 RUN
270 :
280 GOSUB1000:PRINT:PRINT"ATTENDERE...":PRINT:W=0
290 READB:IFB>=0THENPOKET,B:T=T+1:W=W+B:GOTO290
300 READB:IFB<>WTHENPRINT:PRINTCHR$(18)"ERRORE DI TRASCRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X"A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PRINT
330 X1=INT(X/256):X2=X-(X1*256):PRINT"POKE44,"X1":POKE43,"X2;
340 Y1=INT(T/256):Y2=T-(Y1*256):PRINT":POKE46,"Y1":POKE45,"Y2:END
1000 PRINTCHR$(147);
1010 PRINT "STAMPA CARATTERI CUBITALI":PRINT
1020 PRINT "USO: SYS XXXX,A$,FX,FY,LM,XLC,CARSET          [,DEVICE]]":PRINT
1030 PRINT "XXXX=INDIRIZZO DI PARTENZA QUALSIASI"
1040 PRINT "A$  =STRINGA DA STAMPARE"
1050 PRINT "FX  =FATTORE DI INGRAND. ORIZZ. (1...60)";
1060 PRINT "FY  =FATTORE DI INGRAND. VERT (0...255          CON 0=256)"
1070 PRINT "LM  =LEFT MARGIN (0...59)"
```

Beh, cari amici, Tom Bang è un po' ottimista : $100, cioè 256 giorni di compleanno sono già un bel numerino da augurare al nostro Primo Giovedini... In ogni caso, eccoci ora giunti alla fine di questo episodio...

ARRIVEDERCI AL NEXT!

```
1080 PRINT "XL    =NUMERO CARATTERI PER RIGA DI TESTO         (1...255)"
1090 PRINT "CARSET =BYTE ALTO DELLA LOCAZ.INIZIALE      SET CARATTERI (0...255)"
1100 PRINT "        FACOLTATIVO: DEFAULT=208"
1110 PRINT "DEVICE =DEVICE NUMBER (0...255)              FACOLTATIVO: DEFAULT=4"
1120 RETURN
1130 DATA 032,253,174,032,158,173,032,130,183,140,252,003,164,034,132,253
1140 DATA 164,035,132,254,032,241,183,142,054,003,202,224,060,144,005,162
1150 DATA 014,108,000,003,032,241,183,142,052,003,032,241,183,142,255,003
1160 DATA 032,241,183,142,057,003,162,208,032,121,000,201,044,208,003,032
1170 DATA 241,183,142,059,003,162,004,032,121,000,201,044,208,003,032,241
1180 DATA 183,169,060,056,237,255,003,144,198,240,196,160,255,237,054,003
1190 DATA 200,176,250,208,001,096,204,057,003,176,003,140,057,003,172,057
1200 DATA 003,240,242,172,252,003,240,237,169,004,160,000,032,186,255,169
1210 DATA 000,032,189,255,032,192,255,162,004,032,201,255,169,008,032,210
1220 DATA 255,169,000,141,253,003,174,052,003,160,000,142,053,003,140,055
1230 DATA 003,174,255,003,240,013,160,008,169,128,032,210,255,136,208,248
1240 DATA 202,208,243,174,057,003,142,058,003,174,253,003,142,254,003,172
1250 DATA 254,003,177,253,201,255,208,004,169,094,208,028,201,192,144,004
1260 DATA 233,128,176,020,201,128,144,004,233,064,176,012,201,096,144,004
1270 DATA 233,032,176,004,201,064,176,240,133,251,169,000,006,251,042,006
1280 DATA 251,042,006,251,042,109,059,003,133,252,169,128,141,019,003,044
1290 DATA 144,153,044,208,186,044,208,139,174,053,003,172,055,003,169,254
1300 DATA 045,014,220,141,014,220,169,251,037,001,133,001,169,064,133,002
1310 DATA 177,251,202,208,004,174,052,003,200,056,045,019,003,208,001,024
1320 DATA 102,002,176,006,192,008,176,247,144,230,102,002,140,056,003,169
1330 DATA 004,005,001,133,001,169,001,013,014,220,141,014,220,172,054,003
1340 DATA 165,002,032,210,255,136,208,248,078,019,003,208,171,238,254,003
1350 DATA 172,254,003,204,252,003,240,005,206,058,003,208,150,169,013,032
1360 DATA 210,255,172,056,003,192,008,144,135,173,254,003,205,252,003,208
1370 DATA 133,169,015,032,210,255,032,204,255,169,004,076,195,255
1380 DATA 1,43386
1390 END
```

ASCII	CODICE SCHERMO
0....63	0....63
64....95	0....31
96....127	64....95
128....191	64....127
192....223	64....95
224....254	96....126
255	94

Tabella di conversione usata nel programma "Caratteri cubitali"

CONTXL = contatore alla rovescia per lunghezza riga. Quando arriva a zero la stampante va a capo e, se la riga è completa anche in verticale, si pone CNTLEN pari al valore che ha CNTLIN in quel momento.

Notare che se CNTLIN assume valore pari alla lunghezza della stringa, la stampante va a capo indipendentemente dal valore di CONTXL e, terminata l'ultima riga anche in verticale, il programma finalmente termina.

IL DEMO

Il breve programma in Basic proposto come dimostrazione, richiede la routine allocata a partire da $C000 (49152 decimale).

All'inizio chiede la striga da stampare (che per default è "Viva il C/64", vedi riga 235), il byte alto della memoria caratteri e il device. Inoltre domanda se è necessario un controllo per la massima lunghezza di una riga. In caso affermativo, la stringa verrà automaticamente "tagliata" ad una lunghezza pari al massimo valore consentito di XL.

Quindi si passa alla fase di input. I principali parametri sono indicati in alto sullo schermo, e si possono modificare premendo uno dei tasti funzione:

FX:
tasto funzione F1 = +1 unità
tasto funzione F2 = -1 unità

FY:
tasto funzione F3 = +1
tasto funzione F4 = -1

LM:
tasto funzione F5 = +1
tasto funzione F6 = -1

Il tasto F7 manda in stampa la stringa B$. Durante la fase di input dei caratteri vengono accettati solo quelli alfabetici e pochi altri. La stringa è sempre visualizzata sullo schermo, in reverse, poco al di sotto degli stessi parametri.

Vuoi pubblicare un annuncio gratuito?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:

Commodore Computer Club - Viale Famagosta, 75 - 20142 Milano

Acquisterei libretti istruzione Systems per dischetti: "Analisi di Bilancio" - "System Assembler" - "La Voce 3" - nonchè i n. 2-3-5 della rivista "Personal Computer. Telefonare a Iudicello 02 - 4236679 per accordi.

Cerca/offre software e listati per Commodore 64 (oltre 2000 titoli). Rivolgersi Assoc.I.T - Via Papio n.14 - 84100 Salerno

Vendo dischi 5 1/4 per C64 pieni - A/B - a L. 1.000 contenenti giochi e altro vendo anche digit video per C64.
Andrea Iori - Pavullo (MO) - 0536/20665

Vendo/scambio giochi e programmi per C64 su disco e cassetta.
Scrivete (inviando le vostre liste) a Gianluca Pulina - Via Repubblica n. 24 - 27046 S. Giuletta (PV)

Cerco per C128 compilatori Turbo Pascal e Fotran anche in modalità CPM, non in modo 64.
Farabegoli Riccardo - Tel. 040/729626

Cerco programmi per C64 ed in particolare il programma di Totocalcio e grafica per videotitoli.
Scrivere inviando lista a Cirillo Luigi - Via Ghiaie n. 20 - Trento

Per C64 PRG per telecomunicazioni radio o altro vendo su 300 dischetti per videocatalogo spedire n.1 dischetto + 2.000 (grazie).
Samannà Giovanni - Via Manzoni n. 24 - 21027 Paceco (TP) - Tel. 0293/882848

Se hai un'Ammiga, oppure stai per comprarlo e non hai amici con cui scambiare le tue idee o il tuo software.
Scrivi ad Alfonso Roselli - Via Sannio n. 32 - 65016 Montesilvano (PE)

Vendo giochi per C64.
Scrivere o telefonare a Fabrizio Pallotti - 00049 Velletri (RM) - Tel. 06/9631731

Scambio o vendo (a modici prezzi) programmi per C64 e per C128 (modo 128 anche 80 colonne) anche scambio idee per programmazione.
Max serietà.
Telefonare ore serali allo 011/533820

Offertissima!!! Causa realizzo vendo computer perfettamente funzionante SHARP MZ 700 (CPU Z80-64KRam) con registratore incorporato, completo di cavetti, cassetta Basic e 11 giochi, il tutto a 100.000.
Telefonare allo 06/5283790 chiedendo di Claudio.

Per Amiga scambio software nella provincia di Latina.
Scrivere a Gianetti Paolo - V.le Della Vittoria - 04019 Terracina (LT)

Vendo e scambio software per C64!!!
Ultimissime novità ogni mese a prezzo irrisorio (solo floppy) max serietà.
Per informazioni! Moccagatta Pinuccio - Reg. Roncaglie n. 17 - Rivalta.B (AL) - Tel. 0144/72124 (ore pasti)

Mear Soft Systems vende giochi per C64 e Vic 20 (per C64 Robocop - Batman - Platoon ecc.).
Scrivere a: Mear Soft Systems c/o Mendolicchio Armando - Via G. Acquaviva n. 27 - 71100 Foggia

Per C64 e Amiga scambio e vendo ultime novità - Telefonare allo 095/430850 o scrivere a Torrisi Fabrizio - Via Vincenzo Giuffrida n. 107 - 95100 (CT)
L'annuncio è sempre valido!!

RETE - AMIGA - BBS 300 - 1200 BAUD - ore 22.00 - 7.00 - tutti i giorni - Tel 06 - 894525.

Esperto programmatore fa e vende i programmi Basic di ogni genere per il C128 - C64 su richiesta a 500 il blocco (del disco), su disco o cassetta.
telefonare allo 095/686478 dopo le 16.00, per esporre il progr. desiderato.

Vendo programmi per C64 - Oltre 4.000 programmi con le ultime novità - chiedere lista gratis
Gianni Mazzesi - via Cella n. 329 - 48020 S.Stefano (RA) - Tel. 0544/573529

Vendo Software per C64 su disco o cassetta.
Possibilità di abbonamenti. Max serietà.
Per informazioni scrivere o telefonare a Marinaro Angelo - Via G. Matteotti n.7 - 86100 Campobasso - Tel. 0874/69563

Vendo Software per Amiga 500, giochi e utility recentissimi e ad un prezzo bassissimo.
Contattami: Antonio Cupo - Tel. 0771/25588

Vendo programmi per Amiga. Ultimissime novità dagli U.S.A. Vi aspetto: Antonio Cupo - via Matteotti n.2 - Formia 04023 (Latina)

Cerco utenti Amiga per scambi, consigli e possibile amicizia.
Gianluca truffi - Via Vigna Alta - 20075 Lodi - Tel. 0371/610083

Iscriviti all'HACKER Club!
potrai avere gli ultimissimi programmi per C64-128 & Amiga a .4.500 e 5.000! Arrivi settimanali Tel. 0439/44132 e chiedere di Enrico.

Vendo stampante a margherita Commodore DPS 1101 nuova, completa di due set di caratteri e cartucce di nastro di ricambio, software per Word-Processing, per C64 e C128, a 250.000.
Vendo anche Modem 300 baud con interfaccia per C64/C 128, e software per telecomunicazioni a 70.000.
Mario Campli - Via Serpieri n. 13 - 00197 Roma - Tel. 06/872055 ore serali. Si spedisce contrassegni in tutta Italia.

Iscrivetevi al nuovo servizio di abbonamento mensile con tutte le novità per Amiga 500-1000-2000 del Commodore Club Terontola!!
Telefonare dopo le 19.00 allo 0578/294012 (chiedere di Robert) o allo 0575/67616 (chiedere di Andrea)

Vendo giochi e programmi su disco per C64 a 5.500 (supporto magnetico compreso).
Mongiardini Mario - Via Ballerini n. 31 - 20038 Seregno (MI) - Tel. 0362/229890

Vendo C64 con monitor B/N, Disk Drive, registratore, Joy stick, copri tastiera, interfaccia Centronics per 600.000.
Tutto come nuovo. Tel. allo 06/6211834 (ore di cena).

Compro Espansione di memoria per C 128 solo Commodore 1750.
Contattare - Riva Giancarlo - Via G. Verdi n. 1 - Velate Milanese - MI - Tel. 039/672289

Cerco Print Master + Geos 64 + Print Shop + News Room, tutti originali a 60.000 trattabili.
Luca Mineo - Via S.M. Mazzarello n. 5 - 90135 Palermo - Tel. 091/554895

Vendo e Scambio programmi e giochi su C64/C 128.
Segreteria tel.,archivio , condominio, contabilità casa. Giochi: Goblin, Barbarian ecc.
Scrivete o telefonate a Fortino Leonardo - Via Scesa Valle n. 7 - Cabiati (CS) - Tel. 0983/91317 ore:14.00 - 21.50

Causa passaggio a sistema Superiore Vendo C 128 + Drive 1541 + Modem + Mouse + Joystick + Light pen + Manuale italiano + vari floppy e cassette gioco.
Dispongo anche di VIC 20 + registratore + Manuale italiano + Enciclopedia su cassette.
Scrivere a Dalla Libera G.Luca - Via Trieste n. 38 - Collegno (TO) o Telefonare nelle ore serali allo 011/789710

Ti interessi di videogames, bene! Io ho tutte le novità sul Mercato giochi solo su cassetta per C64/C 128.
Francesco Lungangeli - Via De Angeli n. 100 - (RO)
Telefonare alla sera allo 06/434626

Vendo PC 128 Olivetti Prodest completo di Penna Ottica, sette cassette giochi e sei di programmi (contabilità, File Manager ecc.), 2 libri di programmi, a 350.000 trattabili. Causa cambio Sistema.
Telefona allo 02/5513505 e chiedi di Francesco.

Acquisto C64 o C 128 con Disk Drive al prezzo di ca. 350.000.
Mi chiamo Cultrara Roberto - Via Settefontane n. 12 - 34100 Trieste - Tel. 040/750749

Vendo Software vario per C64 - P.Master - P. Shop - News Room - Geos - Easy script - Commodisk dal nx 4 al nx 28 - Simon Basic - Gwbasic - Corso Video Basic 10 dischi origin. - altre utility - giochi etc. in blocco o separ.
Tel. allo 0864/83170 e chiedi di Walter dalle ore 21.00 alle ore 23.00

Cerco "Dilan Dog" con istruzioni (su disco)
Luca Vignale - Via Torino n. 233 - 10032 brandizzo (TO)

Vendo C64 + Drive 1541 + Registratore + portadischi + 80 dischi + 40 cassette - ottime condizioni - nuovi 600.000.
Cappelli Stefano - Via F.Gregorovius n. 30 - Perugia - Telefonare ore pasti allo 075/74493

Cerco Pen Friend (M/F), incluso scambio conoscenze e aiuto applicazioni (C64).
Mauro Marchetti (19 anni) - Via I Maggio n. 36 - 60019 - Senigallia

Vendo Atari (completo di trasformatore e cavo antenna) + 20 giochi c.ca a L. 70.000 è un affare....
Andrea 059/692861 - Telefonare nelle ore pomeridiane.

Heavy Software Club
Cerco possessori di C64/128 pèr formare un club di software.
Via Gennargentu, 15 - 08010 Borore (NU) - Tel. 0785/86640.
Offro/cerco software.
Lauro Lai Francesco - Via Gennergentu, 15 - 08010 Borore (NU) - Tel. 0785/86640.

Vendo /scambio programmi per C64, per avere la lista telefonare allo 0975/23406 e chiedere di Erminio.

Spectrum/MSX: cedo numeroso materiale (programmi, manuali, accessori, libri, riviste, fotocopie italiano, inglese) per cessato interesse.
Telefonare ore pasti al 6881180 e chiedere di Tulli.
Solo zona Milano.

Favoloso: vendo Cartridge Niky 1 originale con relativo manuale in italiano a sole L. 40.000.
Simona - Tel. 02/2566772 (dopo le ore 19,30).

Vendo ultimissimi giochi C64 a prezzi incredibili in tutta italia.
Mario Crisculo - Via Lucatelli, 35 - 00159 Roma - Tel. 4390265.

Vendo e scambio programmi per C64 a L. 3.000 a disco per ricevere lista - Tel. 0975/23406 (ore paura).

Vendo Easy-View + Digi-View-Pal + Digitizer a non meno di L. 100.000.
Luca Costa - Viale dello Sport, 19 - 36031 Dueville (VI) - Tel. 0444/591416 (ore serali).

Porry Software Mail. Vendita video games e programmi per C64.
Ultime novità, prezzi bassi.
Diso/cassetta per lista gratuita Telefonare allo 0535/31009 dalle ore 19,00 alle 22,00.
Chiedere di Flavio.

Vendo tastiera Philips "C7420 Videopac+ " più 4 cartucce, 2 joystik e manuali
di istruzione a L. 150.000 trattabili.
Telefonare a Michele allo 081/8317155.
Scambio programmi per Amiga.

Vendo: giochi e programmi per Amiga 500,con disco L. 5.000, senza L. 3.000.
Mariano Alessio - Via Baldissera, 5/3 - 30174 Mestre (VE) - Tel. 041/5341727.

Scrivo e vendo progr. basic per C 128 (archivi, giochi, ecc.) a L. 500 il blocco (255 bytes) + L. 4.000 per nastro/disco e spese di spedizione.
Chiamate Alessandro dopo le 16,00 allo 095/686478 Biancavilla (CT).

CBM 128 + monitor 1701 + drive 1570 + stampante MPS-803 con trattore + Modem 6499 + tape + penna ottica professionale + 8 portadischi + oltre 300 dischi ds/dd (oltre 2000 prg.) + 50 cassette + 1 slot (tutto con istruzioni).
Perfette condizioni, qualsiasi prova L. 1.900.000.
Francesco Sposato - Via Prati, 114 - Roma - 06/8107337.

Per C64 scambio 5.000 programmi.
Cerco Linker automatico per unire più programmi in un unico file senza uso manuale monitor.
Cerco nuovo catalogo dischi con caricamento automatico Directory.
Sammanà Giovanni - Via Manzoni, 24 - 91027 Paceco (TP).

Ho creato il time cruncher collection che occupa un lato di disco. Lo vendo oppure lo scambio con altro software.
Gori Eugenio - Corso Italia, 48 - 57027 San Vincenzo (Li) - Tel. 0565/703494 o 703977

Vendo penna ottica per C64 a L. 25.000, con software ed istruzioni in italiano.
Vendo inoltre programmi di ogni genere per C64, C128 sia in modalità C128 che in CP/M a L. 500 (disco o cassetta - min. 20).
Giuliano Devicienti - Via T.Ugo Granafei, 5 - 72023 Mesagne (Br) - Tel. 0831/325707.

Cerco espansioni di memoria e accessori per VIC 20.
Telefonare ore serali allo 02/9373944 - Marco.

Cerco possessore di A.500 che mi desse informazioni.
Lorenzi Andrea - Via Lombardia, 30 - 25016 Ghedi (Bs) - Tel. 030/902082.

Cerco amici C16/Plus 4 per scambiare software ed esperienze.
Cosimo Rizzo - P.zza A.Moro, 1 - 73015 Salice S. (Lecce) - Tel. 0832/732386.

Fermi tutti; scommetto che stavate pensando proprio a questo; vi interessano programmi per C64 e Amiga di recentissima "Acquisizione" provenienti dalle Software House americane ed inglesi? Benissimo, se è così non leggete altri annunci e guardate questi prezzi: da L. 2.500 a L. 3.500 (disco compreso) sta per programmi per C64 che per Amiga. Ci sono degli sconti per chi acquista più programmi.
Agli interessati, telefonare a: Fossati Luca - Via Barro,31 - 28045 Invorio (No) - Tel. 0322/55688.

Il Master Soft Club offre ai soci migliaia di programmi di ogni genere su disco o su cassetta per C64/128 e Amiga, bollettino mensile, trucchi, novità, scambi, hardware vario ecc. ecc.
Iscrizione gratuita e massima serietà assicurata. Nicola Gianno - Via Marsala, 351 - 91020 Rilievo (Tp) - Tel. 0923/864559.

Cerco qualsiasi programma attinente a "Koala Painter ".
Scrivete per accordi a: Massimiliano Monaco - Via Don Emanuele Cattaneo, 2 - 20025 Legnano (Mi).

Offro compenso a chi mi fornisce informazioni su stampante STAR SG-15 - Tel. 06/2586627.

FONTI E DIAGRAMMI A TORTA

Due brevi routine per creare grafici a torta e per richiamare i font del WorkBench attraverso le librerie del Kernel di Amiga

Gli Amici di Amiga

Questo mese presentiamo ancora un paio di sottoprogrammi, da "collezione", sviluppati in AmigaBasic. Il primo è un generatore di diagrammi a torta molto semplice e pratico, mentre il secondo è decisamente più complesso e serve a richiamare vari font-caratteri in modo da poterli utilizzare anche in nostri programmi Basic.

PIEGFX

L'insolito nome di questa routine sta per "PieGraphics" (= grafici a torta). Un grafico "a torta" serve per visualizzare graficamente un gruppo di valori sotto forma di fette di una torta: a valore più grande corrisponde una fetta proporzionalmente più grande della stessa torta che rappresenta, quindi, l'unità intera o, se preferite, la somma di tutti i valori.

Per generare diagrammi di questo tipo siamo spesso costretti a ricorrere all'uso di potenti package dedicati al trattamento di dati anche in forma grafica (i cosiddetti spreadsheet o fogli elettronici), come ad esempio il celebre MaxiPlan per Amiga o il Lotus 1-2-3 per i sistemi MS-DOS. Con i programmi citati possiamo realizzare grafici a barre, a torta, a linee, a punti e chi più ne ha più ne metta.

MaxiPlan della Oxxi permette addirittura di creare grafici tridimensionali di ogni tipo, con una semplicità che ha dell'incredibile. Se però interessa un grafico a torta all'interno di un nostro programma Basic? Ecco quindi che entra in scena PieGfx.

UTILIZZO

Come abbiamo già spiegato molte volte, un sottoprogramma è strutturato in modo da essere totalmente indipendente dal resto del programma. Per invocarlo, e passargli i parametri necessari, abbiamo a disposizione l'istruzione CALL. I parametri necessari per attivare correttamente PieGfx sono:

- il numero di valori che desideriamo rappresentare;
- i valori stessi immagazzinati in un array;
- il raggio della torta;
- le coordinate x e y per posizionare la torta;
- la risoluzione utilizzata.

L'array deve immagazzinare, in ogni suo elemento, uno dei valori da rappresentare. Il raggio può assumere qualunque valore che non sia minore di 2. Per quanto riguarda invece la risoluzione dobbiamo indicare

L'ENCICLOPEDIA DI ROUTINE IN AMIGABASIC

Tra le tante possibilità offerte dall'AmigaBasic esiste anche quella di creare sottoprogrammi che sono un insieme di istruzioni, atte a svolgere una particolare funzione, richiamabili da qualsiasi parte del programma. Tale meccanismo di richiamo evita di scrivere la stessa sequenza di istruzioni ogni volta che serve quella particolare funzione.

Le subroutine, che si attivano con un GOSUB, terminano con un RETURN, in modo che il controllo passi alla istruzione successiva al Gosub chiamante.

Il meccanismo di richiamo è valido anche per i sottoprogrammi (a parte le parole-chiave) ma la grande differenza risiede nella gestione delle variabili. Come dice il nome stesso, infatti, un sottoprogramma è un programa a sè stante con le proprie variabili, area di memoria e così via, ma si trova esso stesso all'interno di un programma. Ciò significa che possiamo organizzare le variabili all'interno di un sottoprogramma in maniera molto libera: tenendo conto delle variabili esterne, non tenendone conto affatto, oppure tenendone conto solo parzialmente. La procedura si concretizza nella più assoluta indipendenza del sottoprograma che, quindi, ben si presta per realizzare routine di utilità generale richiamabili da vari programmi. Vediamo ora come si dichiara un sottoprogramma:

```
SUB Nome Sub Program (parametri) STATIC
Istruzioni del sottoprogramma
END SUB
```

Esso può assumere una posizione qualsiasi (ma preferibilmente al fondo) nel programma in quanto non potrà mai essere coinvolto nel flusso di esecuzione: viene semplicemente ignorato, a meno che non facciamo una esplicita richiesta con CALL che, in questo frangente, ha la seguente sintassi...

```
CALL Nome Sub Program (parametri)
```

L'unico "ponte" di comunicazione tra programma e sottoprogramma è rappresentato dai parametri, vale a dire le variabili (o costanti) che intendiamo "passare". Se dichiariamo un certo numero di variabili come parametri in entrata, dovremo effettivamente fornire quello stesso numero di variabili quando effettuiamo la chiamata (CALL).

Molto importante, inoltre, è il "tipo" di variabili passate: non è assolutamente concesso di passare stringhe dove sono richiesti numeri interi, ma neanche passare variabil in floating point (virgola mobile) per interi e così via. Questo vale anche per le costanti: se abbiamo dichiarato le variabili riceventi come f.p. (quindi quelle di default) dovremo passare costanti seguite dal punto esclamativo (!) che denota, appunto, grandezze in floating point.

La parola chiave STATIC indica che le variabili del sottoprogramma sono trattate "staticamente", cioè conservano il loro valore all'esterno del programma pur senza influenzare le eventuali omonime. Potremo accedere nuovamente al loro contenuto solo dopo essere rientrati nel sottoprogramma.

L'Enciclopedia di routine in AmigaBasic ha preso inizio sul n. 58 di Commodore Computer Club e si prefigge di offrire periodicamente numerosi sottoprogrammi di utilità generale.

Coloro che intendono contribuire alla divulgazione di routine di un certo interesse, possono telefonarci per proporne la pubblicazione.

un valore pari a uno (se utilizziamo la bassa risoluzione non interlacciata, 320 x 256, oppure l'alta risoluzione interlacciata 640 x 512).

Bisognerà invece indicare due per l'alta risoluzione non interlacciata (640 x 256). La necessità di disporre di questi valori deriva dal fatto che mentre in bassa risoluzione, oppure in alta interlacciata, i pixel sono quasi quadrati (il rapporto fra le loro dimensioni è di 10 : 11) e quindi i cerchi sono "veri" cerchi, in alta risoluzione non interlacciata i pixel sono spiccatamente rettangolari ed i cerchi diventano ellissi. Inserendo il fattore di raddoppio della larghezza del cerchio riusciamo ad ottenere circonferenze anche in questo modo grafico.

Nel listato pubblicato è inserito il sottoprogramma ed un esempio per illustrare l'uso di PieGfx.

MODIFICHE

Naturalmente nulla vieta di personalizzare la routine nei modi più disparati. Così com'è, infatti, è piuttosto "rozza" in quanto non identifica con diversi colori le varie fette di torta. Al posto dei colori si possono usare anche pattern di riempimento diversi, come abbiamo imparato in precedenza. E' possibile poi aggiungere una legenda di riferimento che associ al colore della fetta il valore rappresentato. Insomma potete abbellire il sottoprogramma come più vi pare, o potete anche lasciarlo ed utilizzarlo così com'è.

La tecnica usata per disegnare l'ampieza delle fette in base al valore rappresentato consiste nella risoluzione della semplice proporzione...

6.2831 : x = TotVal : Arg(x)

...dove TotVal è la somma di tutti i valori, 6.28 è l'intera circonferenza, espressa in radianti, e Arg(x) è il valore preso in esame.

FONT

Questo sottoprogramma, come preannunciato, è piuttosto complesso, non tanto per la funzione che svolge, ma per il fatto che si interfaccia direttamente con le routine del kernel di Amiga. Vediamo prima in dettaglio a che serve.

Nel disco del Workbench esiste una directory denominata "fonts" in cui sono contenuti vari font (tipi di carattere) che possono essere visualizzati da Amiga. I font sono gli stessi che utilizziamo attraverso il programma Deluxe Paint II. Il sottoprogramma Font, riportato in queste pagine, provvede appunto a richiamare, in Basic, il tipo di carattere a cui siamo interessati, in modo da scrivere un testo sul video adot-

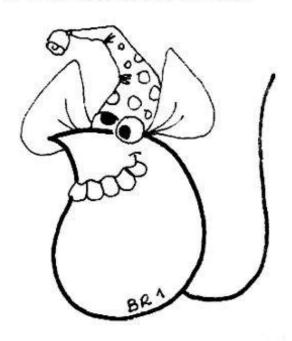
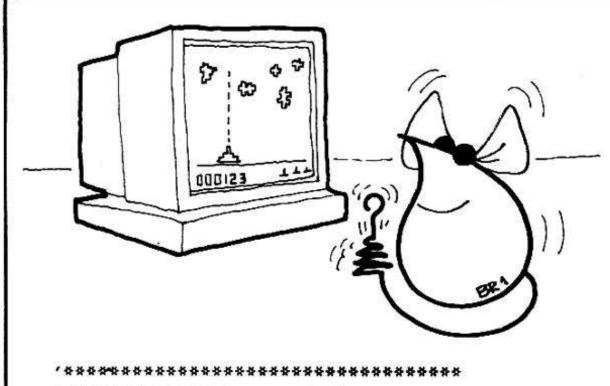
FONT	ALTEZZE		
Ruby	8	12	15
Opal	9	12	
Diamond	12	20	
Garnet	9	16	
Topaz	8	9	11
Emerald	17	20	
Sapphire	14	19	

Altezze disponibili, su Amiga, per ciascun font.

tando il font prescelto.

I parametri da passare al momento del richiamo della routine sono due: il nome del font e la sua altezza. Il nome del font va passato come stringa, l'altezza come numero intero. L'altezza dei caratteri, come ovvio, va decisa in base a quelle realmente disponibili sul disco. Se da CLI chiediamo il contenuto della directory fonts possiamo accorgerci che esistono altre sottodirectory, ognuna con il nome di un font. All'interno di queste sottodirectory troviamo dei file il cui nome è un semplice numero che rappresenta, in effetti, le altezze realmente disponibili per quel determinato font.

Nella tabella 1 è riportato l'elenco completo delle altezze disponibili.

Il programma mostra chiaramente che il richiamo di un sottoprogramma può avvenire anche senza la parola chiave CALL: è sufficiente il nome e la lista dei parametri. L'unico caso in cui è obbligatorio specificare CALL è dopo un THEN o un ELSE, in modo da distinguere se si tratta di una chiamata ad un sottoprogramma oppure di un semplice salto ad una label.

```
'***********************************
' SottoProgramma .... Font
' Linguaggio ........ AmigaBasic
' Autore ...........
'-----------------------------------
' Richiama font diversi sfruttando
' le librerie di sistema (library)
'***********************************

DECLARE FUNCTION OpenDiskFont& LIBRARY

LIBRARY "diskfont.library"
LIBRARY "graphics.library"

Font "garnet.font",16
LOCATE 1,12:PRINT "I Fonts di Amiga !"
Font "ruby.font",15
LOCATE 3,11:PRINT "Questo è il ruby..."
Font "sapphire.font",19
LOCATE 4,11:PRINT "... e questo il Sapphire !"
Font "diamond.font",20
LOCATE 5,11:PRINT "Diamond 20 by Amiga"
Font "topaz.font",8

LIBRARY CLOSE
END

SUB Font(nomeFont$, altezza%) STATIC
  IF pFont&<>0 THEN CALL CloseFont(pFont&)
  nomeFont$ = nomeFont$+CHR$(0)
  attrFont&(0) = SADD(nomeFont$)
  attrFont&(1) = altezza%*65536&
  pFont&=OpenDiskFont&(VARPTR(attrFont&(0)))
  IF pFont& <> 0 THEN SetFont WINDOW(8),pFont&
END SUB
```

```
' ************************************
'  SottoProgramma .... PieGfx
'  Linguaggio ........ AmigaBasic
'  Autore ...........
' ------------------------------------
'  Esempio d'utilizzo del subprogram
'  PieGfx per creare grafici a torta
' ************************************

Par=6
V(1)=1000:V(2)=2000
V(3)=5000:V(4)=4500
V(5)=4000:V(6)=500
CALL PieGfx(Par,V(),60!,350!,100!,2!)
END

SUB PieGfx(Na,Arg(Par),Rg,xx,yy,Ris) STATIC
 IF Ris<1 OR Ris>2 OR Rg<2 THEN
  BEEP:EXIT SUB
 END IF
 FOR i=1 TO Na
   TotVal=TotVal+Arg(i)
 NEXT i
 PSET (xx+Rg*Ris,yy),2
 FOR i=1 TO Na
   ar=(Arg(i)/TotVal)*6.283185
   FOR a=ar1 TO (ar1+ar) STEP .04
     x=COS(a):y=-SIN(a)
     LINE -(x*Rg*Ris+xx,y*Rg+yy),2
   NEXT a
   LINE (xx,yy)-(x*Rg*Ris+xx,y*Rg+yy),2
   ar1=ar1+ar
 NEXT i
END SUB
```

LE LIBRERIE

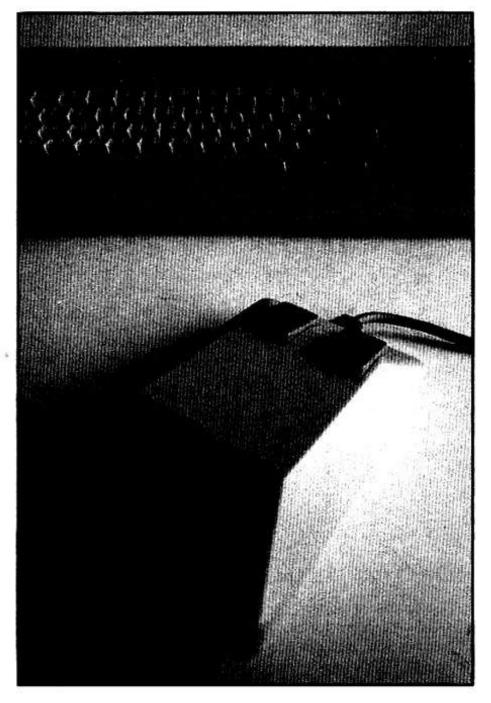
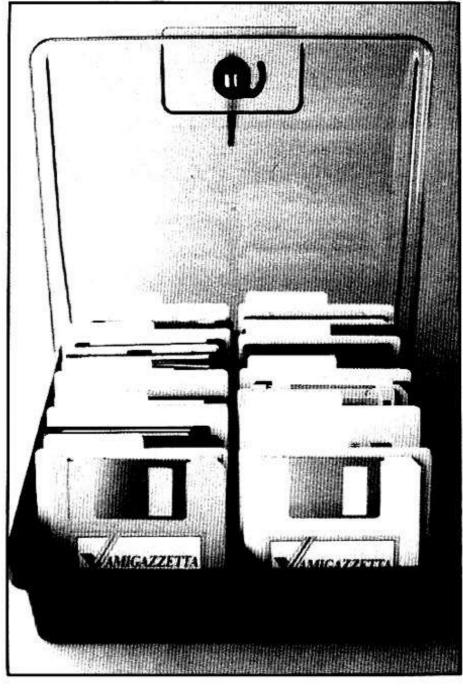
Font utilizza le funzioni del Kernel di Amiga. Per accedere ad una determinata funzione è però necessario "aprire" la libreria che la contiene. Nel nostro caso utilizziamo le funzioni OpenDiskFont(), SetFont() e CloseFont(). La prima è contenuta nella libreria "diskfont.library", la seconda e la terza nella libreria "graphics.library". Per aprire una libreria da AmigaBasic ci serviamo dell'istruzione LIBRARY, seguita dal nome della libreria.

Per usare una funzione è anche bene dichiarare il tipo di valore restituito dalla stessa. In questo caso ci serviamo di DECLARE FUNCTION funzione LIBRARY, dove "funzione" è il nome della funzione seguita dal carattere che ne identifica il valore di ritorno (nel caso di OpenDiskFont c'è la E commerciale (&) che indica un valore intero a 32 bit).

Per aprire una libreria, però, non è sufficiente la sola istruzione LIBRARY, occorre creare sul disco in cui abbiamo registrato il programma, un file detto "file .bmap", uno per ogni libreria a cui intendiamo accedere. Nel nostro caso serviranno dunque i file .bmap per le librerie diskfont e graphics. Il file "graphics.bmap" lo possiamo rintracciare nella directory BasicDemos del disco Extras e pertanto sarà sufficiente ricopiarlo sul nostro disco. Il file diskfont.bmap, invece, dobbiamo crearcelo seguendo questa procedura:

- Avviare Amiga con il disco del Workbench ed inserire il dischetto Extras;
- Aprire la directory BasicDemos di Extras e lanciare il programma ConvertFD;
- Rispondere con "Extras:FD1.2/diskfont-lib.fd" alla richiesta del file .fd:
- Rispondere con "nomevostrodisco:diskfont.bmap" alla richiesta del file .bmap;
- Attendere alcuni secondi in modo che il programma crei sul vostro disco il file diskfont.bmap.

Se avete seguito alla lettera la procedura dovreste avere tutto l'occorrente: non vi resta che lanciare il programma dimostrativo ed osservarne i risultati.

Prima di lasciarci terrei a precisare che una routine simile a questa presentata la potete trovare nel programma "Library" facente parte alla directory BasicDemos del disco Extras. La sostanziale differenza consiste nell'effetto che ognuna ha sui caratteri. Mentre quella contenuta nel programma Library cambia solo lo stile dei caratteri, Font cambia proprio il tipo di carattere, non lo stile.

FORSE NON TUTTI SANNO CHE...

Quasi certamente ritenete di sapere tutto su Amiga. E' il momento di dimostrarlo confrontando le vostre risposte con quelle esatte

1) ORDINE ALFABETICO

Si supponga di avere in Ram Disk un file di testo (ciò contenente caratteri in formato Ascii) di nome *Rubrica*, contenente le seguenti stringhe...

Mario Rossi*
Ugo Bianchi*
Sergio Verdi*

... con l'asterisco posto in fondo ad indicare un *line feed*, ovvero la conclusione di ogni singola stringa.
Dire con quale comando del Dos, nella sua corretta sintassi, è possibile avere una stampa su carta del file, con il suo contenuto ordinato alfabeticamente.

2) ANCORA ORDINE

Volendo ordinare il file di cui al punto 1 per cognomi, con quale sintassi andrebbe impartito lo stesso comando, ed in quale sequenza verrebbero stampati i 3 nominativi?

3) DIMMI QUANTO VALE

Dire quale valore verrà assegnato alla variabile c%, mandando in esecuzione il seguente programma in AmigaBasic:

```
a% = 4
b% = 2.5
c% = a% * b%
PRINT c%
```

4) IMMAGINI E NON

Quando di un file si dice che è in formato IFF, significa che si tratta di una schermata grafica?

5) UNA DIRECTORY PER STAMPARE

Elencare esattamente quali files devono essere presenti in un dischetto per abilitare all'uso una stampante, citando anche la directory che deve contenerli.

6) UN ELENCO "SPECIALE"

Supponendo che il sistema sia stato attivato con il disco Workbench, e che lo stesso sia la directory corrente, che tipo di output si otterrà dopo il comando *List :C Quick Keys*

7) BATCH CHIAMA BATCH

Può un File Batch richiamarne un altro, inserendo nel suo contesto un comando *Execute*?

8) INPUT

Esiste, adoperando esclusivamente i comandi del Dos, la possibilità di ottenere un *input* da tastiera? E, in caso affermativo, tramite quale comando?

9) DUE FINESTRE

Si immagini di aver aperto una finestra Cli dal Workbench, e digitato al suo interno il comando *Cd C*, seguito poi da un *Newcli*, per accedere ad una seconda finestra Dos.
Trovandosi in quest'ultima, viene mantenuta la stessa assegnazione di directory corrente (cioè *c*)?

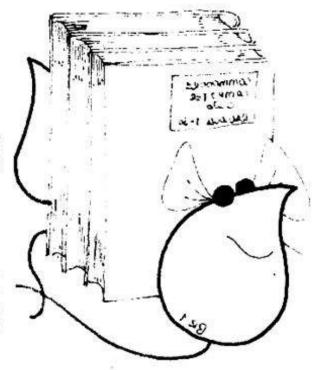
10) UNA... PORTA FINESTRA

Con quale sintassi del comando NewCli è possibile configurare una nuova finestra in modo che occupi tutto lo schermo visibile?

11) COLORI IN BASIC

Con quali parametri va settato il comando Basic *Palette*, per impostare il colore bianco, e quali per il nero?

12) SCROLLARE O NON SCROLLARE

Si consideri una finestra occupante tutto lo schermo sul quale è aperta. Tale schermo, in ambiente Basic, è stato definito con una risoluzione orizzontale di 640 pixel, ovvero con la possibilità di visualizzare 80 caratteri.

Dite che cosa avverrà se si stampa una stringa composta da 100 caratteri, per esempio con...

```
PRINT STRING$ (100, "*")
```

13) MAIUSCOLO E MINUSCOLO

Si esamini il seguente programma basic...

```
attesa:
a$ = INKEY$
IF a$<>"s" THEN GOTO attesa
PRINT "fine": END
```

...che attende la pressione di un tasto, per poi stampare la stringa "fine". Il ciclo di attesa verrà superato solo se il tasto corrisponde ad "s".

Sapreste dire se digitandolo in maiuscolo (con Shift oppure Caps Lock) si otterrebbe lo stesso risultato?

14) ERRORE

Dopo un ipotetico Run a questo listato...

```
SCREEN 2, 320, 250, 2, 1
WINDOW 3, , , ,2
PALETTE 4, 1, 1, 1
COLOR 3, 4
 PRINT "premi un tasto"
10 a$ = INKEY$
 IF a$= "" THEN 10
SCREEN CLOSE 2
```

...verrà segnalato un errore *Illegal Function Call*.

Dite perchè, ed a quale istruzione del listato fa riferimento.

15) DOPPIO ERRORE

Cosa è accaduto, se AmigaBasic segnala un errore del tipo *Duplicate Definition*?

16) FILES

Volendo leggere una serie di dati da un file (su disco o Ram), senza conoscerne il numero, a quale funzione del Basic bisogna ricorrere?

17) VALORI

Quali valori numerici stamperà il programma che segue?

```
a=5: b=4
calcola a, b
PRINT x; y; z
END
SUB calcola (x, y) STATIC
SHARED z
z = x*y
END SUB
```

18) TAGLIA E INCOLLA

Quando, con AmigaBasic, si usa l' opzione *Cut* del menu *Edit*, dove vanno a finire le parti di programma eliminate dal listato?

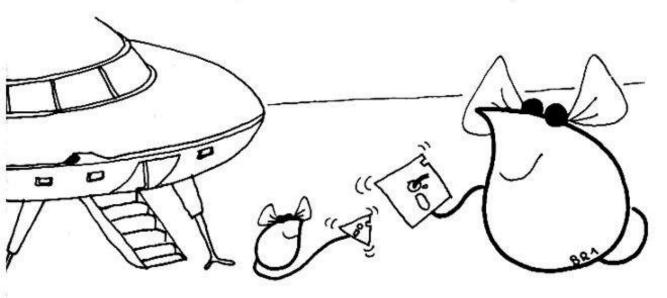
19) ELIMINARE LE ICONE

Dite con quale comando del Dos è possibile cancellare, dalla directory corrente, tutti i file con suffisso ".info", ovvero le icone.

20) SENZA CARATTERE

Com'è noto, nel disco Workbench è presente la directory *Fonts*, contenente vari tipi di carattere che possono essere utilizzati da Amiga.

Eliminando la directory (ed il suo contenuto) dal dischetto, il sistema è in grado di funzionare correttamente?

RISPOSTE AL QUIZ SU AMIGA

1) ORDINE ALFABETICO

Il comando del Dos, con cui avere una stampa su carta del file *Rubrica* è:
Sort Ram:rubrica prt:

2) ANCORA ORDINE

Alla sintassi della precedente risposta occorre aggiungere *Colstart*, che stabilisce da quale colonna iniziare l'ordinamento.

Le stringhe proposte nella domanda, sono conformate in modo che, con opportune spaziature, i cognomi inizino tutti alla ottava colonna.

Quindi, basterà impartire...
Sort Ram:rubrica prt: colstart 8
...per ottenere un output su carta così formato:
Ugo Bianchi
?ario Rossi
Sergio Verdi

3) DIMMI QUANTO VALE

La moltiplicazione a% * b% darà come risultato 8, assegnandolo alla variabile c%.

In effetti il prodotto di 4 * 2.5 è uguale a 10, ma, attribuendo alla variabile b le caratteristiche di valore intero, tramite il simbolo di percentuale (%), la relativa assegnazione troncherà la parte decimale di 2.5, lasciandone solo la parte intera (2).

In definitiva l'operazione avverrà tra 4 (variabile a%) e 2 (b%).

4) IMMAGINI E NON

Se un file è in formato IFF, non necessariamente significa che si tratta di una schermata grafica.

IFF, infatti, significa *Interchange File Format*, e costituisce, in pratica, uno standard per memorizzare dati di qualunque tipo.

Utilizzando lo stesso tipo di struttura, i dati possono così essere sfruttati dal software più svariato.

L'esempio più lampante sono proprio le schermate grafiche, ma lo standard può essere applicato a files contenenti colonne sonore, testi, o altro.

5) UNA DIRECTORY PER STAMPARE

I files che devono essere presenti in un dischetto, per abilitare all'uso una stampante, sono 3, e precisamente:

- Port-Handler, contenuto nella directory L.
- Printer.device, inserito nella directory Devs.
- Il file indicante il modello di stampante posseduta, che deve trovarsi nella directory Printers, a sua volta collocata in Devs.

I file sono normalmente rintracciabili nelle stesse directory del disco Workbench, se si dispone della versione 1.2.

Con la release 1.3, invece, il terzo file è situato nel dischetto Extras, con lo stesso "percorso" prima indicato (Devs/Printers).

6) UN ELENCO "SPECIALE"

L'output che si otterrà dopo il comando...
List :C Quick Keys
...sarà l'elenco dei comandi contenuti nella directory C, senza le solite indicazioni concernenti data, bit di protezione e commenti, ma con, al loro posto (tra parentesi quadre), il numero del primo blocco-dati occupato dal file nel dischetto (chiamato File Header).

Molto utile per appassionati "cracker".

7) BATCH CHIAMA BATCH

Un File Batch può richiamarne un altro, inserendo nel suo contesto un comando *Execute*, ma è necessario tener presente che il sistema opera, prima della nuova esecuzione, un backup provvisorio dei comandi del file, inserendoli nella directory T (se non c'è, la crea).

Per questo motivo, a meno che non si stia utilizzando la Ram Disk, è necessario consentire l'accesso in scrittura al disco corrente.

In caso contrario, Amiga avvertirà con un requester, e, se optiamo per "Cancel", la procedura Execute abortirà.

8) INPUT

Anche se non è proprio un input in piena regola, il Dos mette a dispozione la possibilità di effettuare qualche scelta con il comando *Ask*, che può essere seguito da una stringa di caratteri.

La stringa verrà stampata, dopodichè il sistema resta in attesa della pressione di un tasto.

Se questo è uguale a *y* (yes), la pseudo - variabile *Warn* assumerà valore 5, in caso contrario varrà 0.

Per utilizzare tale risorsa, si può ricorrere a condizioni *If ... endif*, per esempio così:

```
Ask CONTINUO?
If not warn
 quit
endif
;continuazione
```

9) DUE FINESTRE

Due finestre di CLI, aperte contemporaneamente, generano due "task", l'uno indipendente dall'altro.

Se infatti, dopo aver clickato nella prima finestra, si impartisce *List*, verrà visualizzato il contenuto della directory *C*.

Ripetando la stessa operazione nella seconda, il List agirà sulla directory principale (Root), sempre che non ci si sia addentrati in altre "ramificazioni" del disco.

10) UNA... PORTA FINESTRA

La sintassi per ottenere la configurazione di una nuova finestra, che occupi tutto lo schermo visibile, dipende dalla risoluzione di schermo scelta tramite le *Preferences*: con 80 colonne (senza interlace), il comando sarà...
Newcli con:0 / 0 / 640 / 250 / Nome finestra
...che per il modo 60 colonne diventerà...
Newcli con:0 / 0 / 320 / 250 / Nomefinestra
Per la *Shell* il dotazione al WB 1.3, valgono gli stessi parametri, solo che *NewCli* diventerà *Newshell*, mentre va usato *Newcon:* al posto di *Con:*.

11) COLORI IN BASIC

I parametri con cui va settato il comando Basic *Palette*, per impostare i colori, sono funzione delle quantità di rosso, verde e blu che definiscono un colore (sempre comprese tra 0 ed 1).

La mancanza di tutte le tonalità, rappresentata dal valore 0 in tutte e tre le percentuali cromatiche, fornirà il nero, mentre

l'opposto (tutte le percentuali al loro valore massimo) produrrà il bianco.

In definitiva, con X da sostituire con un numero identificatore, avremo:

Nero = Palette X, 0, 0, 0
Bianco = Palette X, 1, 1, 1

12) SCROLLARE O NON SCROLLARE

Se si stampa una stringa composta da 100 caratteri, quando lo schermo ne potrebbe ospitare solo 80, verranno visualizzati solo i primi 76 caratteri; gli altri deborderanno al di fuori dello schermo, risultando invisibili.

Se si desidera contenere entro i limiti della finestra di output il "materiale" da stampare, è necessario adoperare il comando *Width X*, con *X* che rappresenta il numero massimo di caratteri consentiti per ogni linea.

Quindi, con...

```
WIDTH 76
PRINT STRING$ (100, "*")
```

...la stringa di 100 asterischi verrà mostrata interamente, anche se su due righe di schermo.

Il parametro X, è intuitivo, va in pratica adeguato alle dimensioni della finestra.

13) MAIUSCOLO E MINUSCOLO

Per soddisfare la condizione di (dis)eguaglianza, occorre che il carattere sia proprio come specificato nel testo del listato, owero *minuscolo*.

E' buona norma, infatti, quando si desidera una risposta da tastiera, adoperare una forma di questo tipo:

```
attesa:
a$ = UCASE$ (INKEY$)
IF a$<> "Y" THEN GOTO attesa
```

In questo caso, il carattere digitato viene trasformato in maiuscolo da *Ucase$* (se non lo è già) e, dato che nella condizione *If* lo stesso è specificato proprio in maiuscolo, praticamente si potrà rispondere da tastiera senza badare alla eventuale pressione di Shift o Caps Lock.

14) ERRORE ILLEGAL FUNCTION CALL

Il parametro "4" dell'istruzione Color, presuppone che possano essere usate 5 palette di colori (da 0 a 4).

Nella definizione delle caratteristiche dello schermo, viene però precisato (quarto parametro di Screen) che esso dispone di 2 bitplanes, e, quindi, può visualizzare un massimo di 4 colori (palette da 0 a 3).

L'errore interverrà, dunque, quando l'interprete tenterà di eseguire *Color*.

E' necessario, volendo lasciare inalterato questo comando, portare a 3 il valore di profondità dello schermo, inserendolo come parametro di Screen.

Non si dimentichi che, così facendo, viene impegnata una maggior quantità di memoria, dato che uno schermo con 2 bitplanes (4 colori) occupa 16 Kbyte, mentre per 8 colori (3 bitplane), necessitano 24 Kbyte.

15) DOPPIO ERRORE

Se AmigaBasic segnala un errore del tipo *Duplicate Definition*, vuol dire che si è tentato di dimensionare una variabile ad indice (vettore o matrice) più di una volta, oppure la variabile era già stata utilizzata prima dell'istruzione Dim.

16) FILES

Volendo leggere una serie di dati da un file bisogna ricorrere alla sintassi Basic EOF(x), owero End Of File, che segnala appunto la fine del file (X = numero del file).

In pratica, dopo la consueta operazione di apertura (Open), basta inserire i comandi di lettura all'interno di un ciclo *While ... Wend*, che controlli lo stato di Eof.

La funzione risulterà uguale a -1 se la fine è stata raggiunta, valore che AmigaBasic interpreta come "vero".

Quindi, con NOT EOF(x), può essere testata la condizione opposta, owero il mancato raggiungimento della fine del file.

Ecco un esempio che legge e stampa su video il contenuto di un file (di tipo Ascii) carattere per carattere:

```
OPEN "nomefile" FOR INPUT AS 1
WHILE NOT EOF(1)
 a$ = INPUT$(1,1)
 PRINT a$;
WEND
CLOSE #1
```

17) VALORI

Il programma riportato nella domanda stamperà, nell'ordine: *0, 0 e 20*.

Le tre variabili usate da Print, infatti, svolgono il loro compito solo nell'ambito della subroutine "cancella".

X ed Y vengono qui dichiarate *Static*, e quindi non assumono alcun valore nella sezione principale del programma.

Z, invece, specificata come *Shared*, manterrà la sua assegnazione tanto nel programma che nella subroutine.

E, dato che il comando *Print x; y; z* si trova al di fuori di quest'ultima, solo la variabile Z risulterà significativa.

X ed Y, nella Subroutine, "ricevono" il valore di A e B.

18) TAGLIA E INCOLLA

Il comando *Cut* "deposita" nella Ram Disk le parti eliminate, come pure le parti selezionate con l'opzione Copy.

La cosa, da Cli, è facilmente verificabile con *List Ram:*, che rileverà la presenza di un file con nome *basicclip*.

Per dargli un'occhiata con Type, è necessario usare la forma...

Type Ram:basicclip opt H

...in quanto, trovandoci in presenza di un file binario, una risposta in Ascii può dare risultati imprevedibili.

19) ELIMINARE LE ICONE

Il comando del Dos, con cui cancellare dalla directory corrente tutti i file con suffisso ".info" (owero le icone) è *Delete*, associato al simbolo #? che includerà tutte le ricorrenze di caratteri che seguono o precedono la specifica del comando.

In pratica, l'esito desiderato si otterrà con:

Delete #?.info.

20) SENZA CARATTERE

Eliminando la directory *Fonts* (ed il suo contenuto) dal dischetto WB, il sistema è in grado di funzionare correttamente, anche se non si potrà modificare il "Topaz" di default della macchina.

Provando, dopo tale eliminazione, ad entrare in Cli ed impartire un comando *Assign*, si constaterà come l'assegnazione di Fonts è lasciata vuota.

Può risultare utile non includere la directory in questione nel proprio disco di lavoro, in modo da guadagnare ulteriore spazio.

Se, poi, qualche applicazione richiedesse l'uso dei font, è sempre possibile rimediare con...

Assign Fonts: Nomedisco:Fonts

...con "nomedisco" che può indicare una seconda periferica (df1:), oppure il nome di un altro dischetto, purchè vi sia memorizzata una directory Fonts.

PER QUALCHE INFORMAZIONE IN PIU'

Il C/128, a giudicare dalle continue richieste di delucidazioni, sembra vivere una seconda giovinezza. Ecco alcuni argomenti di sicuro interesse

di **Domenico Pavone**

LA SOLITA MUSICA

Il numero speciale *Disk'o 'teca* (l'interessante pubblicazione su disco apparsa in edicola che è in grado di eseguire musica con il C/64), ripropone una domanda piuttosto ovvia:

E' possibile far girare, in modo 128, i brani musicali che è possibile ascoltare in modo 64?

I brani musicali contenuti nello speciale *Disk'O 'Teca*, sono tutti realizzati con complesse, e non rilocabili, procedure in linguaggio macchina.

Come se non bastasse, alcuni di essi sono ottenuti con tecniche di campionamento sonoro.

Ergo, una loro conversione per il C/128, pur non impossibile (specialmente per quelli inseriti nella facciata B), richiederebbe un lavoro tale da sconsigliarne la messa in opera, a meno di non possedere una più che approfondita padronanza del linguaggio macchina (nonchè una pazienza certosina).

Per accostarsi all'argomento, comunque, è sufficiente spulciare tra quanto già pubblicato a proposito del C/64, in quanto la gestione del sonoro è *assolutamente* identica nelle due macchine.

Un tipico esempio lo si può trovare nell'inserto *Campus* del n. 63 (pagina VIII), da

cui si possono ricavare, pari pari, tutti gli indirizzi e la struttura dei registri del SID; e, addirittura, applicare al C/128 la routine proposta, modificandone solo la allocazione e l'indirizzo di "ritorno" al normale interrupt.

Più in pratica, ed a patto di (saper) adoperare un editor / assembler, basta modificare la riga 2 del disassemblato in...

IRQSYS = $FA65

...e la riga 7 in...

*=$1300

...mentre, da Basic, l'attivazione diventerà...

Bank 15: Sys 4864

Chiaramente, il listato caricatore proposto in quella sede non risulta più valido per la nuova allocazione.

LE SYSTEM - ROM

Spinti da alcune richieste (che, tuttavia, sono ancora troppo poche... capita l'antifona?) ci stiamo organizzando per illustrare la gestione di alcune Sys-Rom relative al C/128. Ci sia consentita, però, una osservazione: sul numero 56 di C.C.C. è apparso un articolo riguardante il C/128, a pagina X dell'inserto.

Non si tratta di una tabella vera e propria, d'accordo, ma, andando a scorrere il disassemblato ed i suoi commenti (per non parlare del testo), vi si possono ritracciare gli indirizzi di almeno una decina di Rom-routine.

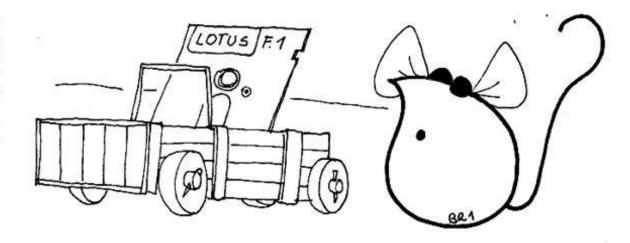
A quell'articolo, nei numeri successivi, sono seguiti parecchi altri: se la matematica non è un'opinione...

FELICITA' E'...

"Vi invio una routine in linguaggio macchina per C/128, in grado di far risuscitare un programma Basic inavvertitamente cancellato da un New o da un Reset. Vi prego di pubblicarlo, mi farete senz'altro felice." scrive il nostro lettore Vincenzo Vullo, riproponendo, indirettamente, le modalità di collaborazione alla nostra rivista.

Il fatto che Vincenzo si avvicini al linguaggio macchina, e con il C/128 per giunta, non può che farci piacere.

Alla sua felicità, però, deve corrispondere anche quella degli altri, nonchè la nostra.

Il listato inviato, infatti, pur composto da solo una ventina di righe comprendenti un centinaio di Data, presenta due pecche:

- Non è accompagnato da alcuna nota o disassemblato, e, soprattutto, viene inviato solo su carta (avviso per tutti i lettori!).

- Del centinaio di byte che costituiscono il codice macchina, circa la metà sono dedicati ad un personale messaggio di copyright (più che legittimo) ma occorre tener presente la pigrizia digitereccia di quasi tutti i divoratori di listati (avviso numero due).

Tolto il messaggio, ed assente la componente didattica, ciò che resta è un efficace *Old*, che comunque può essere facilmente ottenuto anche da Basic, e con minor fatica, semplicemente digitando in modo diretto (da banco 15):

```
a = peek (45) + peek (46) * 256
poke a + 1, 29: sys 20303
```

Il nostro lettore, però, leso nel suo sacro diritto alla felicità, non mediti propositi suicidi: ritentar non nuoce.

GRAFICA IN L.M.

Sul numero 62 di CCC è stata pubblicata la traduzione in linguaggio macchina di quasi tutti i comandi grafici del C/128, ad eccezione di Rdot (x), di cui ci occupamo adesso, spinti soprattutto dalla specifica richiesta del nostro lettore Luca Cassioli di S. Angelo Romano.

La funzione *Rdot (n)*, a seconda del valore espresso dal parametro *n*, restituisce la posizione corrente del cursore pixel, o la sorgente colore cui fa riferimento il comando grafico del Basic (si veda il manuale) impartito per ultimo.

La routine dell'interprete che gestisce questa funzione è allocata a partire dall'indirizzo esadecimale $9B0C (decimale 39692), ma, programmando in Assembly, la sua utilizzazione è decisamente superflua, come già affermato nell'articolo sul n. 62.

All'indirizzo citato, infatti, comincia una procedura che possiamo così schematizzare:

1) Viene prelevato, ed inserito nel registro X, il contenuto del parametro n di Rdot.

2) Se questo è uguale a 0 oppure a 1, viene attivata una sezione della routine che inizia alla locazione $9B23, per leggere le coordinate X ed Y del punto in cui si trova il cursore pixel, prelevandole dalle locazioni $1131 - $1132 (decimale 4401/2) per X e $1133 - $1134 (decimale 4403/4) per Y.

3) Se il registro X, invece, risulta contenere un valore 2, sarà interessata la subroutine di sistema $9C49, adibita a leggere la sorgente colore usata per disegnare, che viene letta dalla locazione $83 (decimale 131).

Dopo tali manovre, i valori ricavati vengono debitamente trasformati per essere letti da Basic, per esempio con Print Rdot(n).

Operando, però, in linguaggio macchina, non è necessario ricorrere a tali routine.

Per ciò che riguarda la sorgente colore, come chiaramente descritto nell'articolo citato (descrizione di *Color*), siamo proprio noi ad inserire il valore desiderato nella locazione $83: non ha senso, quindi, leggerlo attraverso una routine di sistema, e, comunque, un semplice Lda $83 svolge lo stesso compito, ed in maniera decisamente più veloce; da Basic: peek (131).

Inoltre, come forse non chiaramente espresso in quell'articolo, dopo che una routine grafica dell'interprete ha svolto il suo compito (tanto da Basic che da LM), in $1131 - $1132 (coordinata orizzontale) e $1133 - $1134 (coordinata verticale) viene *sempre* a trovarsi (in formato basso / al-

to) l'ultima posizione assunta dal cursore pixel.

Anche qui, dunque, basterà leggere le suddette locazioni, senza ricorrere a Rdot.

Tutte le locazioni ed indirizzi fin qui citati, vanno riferiti alla configurazione di memoria espressa da BanK 15, che va obbligatoriamente specificata se si vuole accedere ad essi tramite il Monitor LM.

Se, per esempio, si vuole vedere il disassemblato della routine Rdot, occorre impartire (da Monitor) un comando D F9B0C, con "F" (= 15 in esadecimale) che sta ad indicare il numero di banco.

Digitando, invece, solo D 9B0C, il Monitor assumerà per default il banco 0, visualizzando una sfilza di punti interrogativi e Brk (= Break) se in quelle locazioni non è memorizzata qualche nostra routine LM o listato Basic.

PRINT SENZA SCROLLING

A beneficio di Roberto Francini di Prato, ma anche dei lettori meno esperti, affrontiamo in questa sede la tecnica di gestione del cosiddetto *scrolling*, riferendoci ad un banale esempio ed operando, inutile ricordarlo, in modo 128.

Si pulisca lo schermo; quindi ci si porti con il cursore sull'ultima riga dello stesso e si digiti una serie di caratteri scelti a caso, senza giungere all'ultima posizione a destra.

Ora, dopo essere tornati alla posizione "home", si digiti il seguente listato...

```
10 for x=1 to 40
20 a$ = a$ + chr$ (215)
30 next x
40 print a$
```

...e si impartisca Run.

Verrà stampata una linea formata da 40 pallini vuoti, ma, come si può constatare, i caratteri in fondo allo schermo saranno scomparsi.

In pratica, tutto ciò che si trovava al di sotto della serie di pallini è stato spostato di una riga verso il basso, con relativa *espulsione* dell'ultima riga.

Il C/128, come è noto, implementa la possibilità di disabilitare lo scrolling verso l'alto tramite la successione dei tasti Escape + M (da programma = chr$(27) + "M"); mentre, per ovviare all'inconveniente appena visto, bisogna intervenire con un artifizio.

Il motivo dello scrolling, infatti, è da ricercarsi nel *link* (collegamento) tra linee fisiche di schermo, per adeguarsi alla struttura logica della linea stessa.

Più chiaramente: una riga logica, com'è noto, può raggiungere i 160 caratteri.

Una volta raggiunti i 40 (operando in modalità 40 colonne), il sistema crea la "continuazione" della linea in modo che possa raggiungere gli 80 caratteri (e così via sino alla eventuale quarta riga), procedendo al cosiddetto *wrap*, che, in ultima analisi, si traduce in uno scrolling verso il basso.

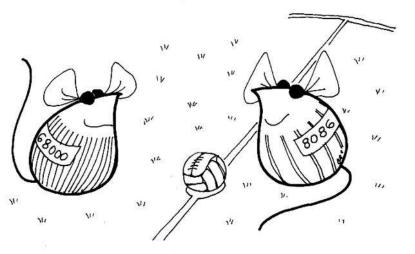
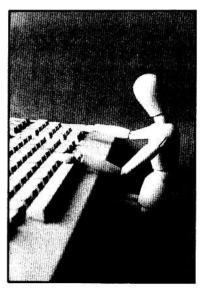
Per far ciò, esamina prima un "flag", ovvero una locazione di pagina zero, posto all'indirizzo $F8 (decimale 248), testandone il bit 6.

Se quest'ultimo è azzerato, il S.O. ne deduce che le linee fisiche dello schermo possono essere "linkate" per formare linee logiche.

Se il bit è invece settato (= 1), il link viene disabilitato.

Quindi, in conclusione, basterà aggiungere al listato appena visto due linee:

```
35 poke 248, peek (248) or 64
```

...per settare il bit 6 del byte $F8, e...

```
45 poke 248, peek (248) and 191
```

...per resettarlo dopo il print.

La stessa locazione, per inciso, controlla anche l'implementazione di Esc + M, che può essere attivata settandone il bit 7.

PROBLEMI DI TURBO

E' un vero peccato che per il C/64 siano presenti parecchie cartucce velocizzatrici, mentre per il C/128... neanche a parlarne. Tale carenza, evidenziata da molti lettori (tra cui G. Fontana di Milano) risulta purtroppo, paradossalmente, giustificata.

Il drive 1541, in associazione ad un C/128, è infatti una soluzione decisamente di ripiego, dato che questo computer è stato progettato in modo da essere collegato ad unità tipo 1570 o 1571.

Tali periferiche, come è noto, sono già predisposte per funzionare a ritmi "turbo", oltre ad implementare caratteristiche esclusive come, ad esempio, il boot automatico di programmi.

Il problema di una maggiore velocità di caricamento potrebbe essere risolto via software, ma, come si può verificare esaminando la routine di fast load per C/64 pubblicata su Speciale Drive, una sua eventuale conversione per C/128 richiederebbe spazi (ed impegno) non certo irrisori.

Il che, per una conformazione di sistema particolare, come quella posseduta da alcuni lettori, non ne giustifica (al momento) la pubblicazione.

Se, però, una valanga di lettere ci convincesse del contrario...

FOTOGRAFA L'ESTATE CON IL COMPUTER

Non è necessario essere programmatori per collaborare con Commodore Computer Club

Vacanze, libertà, eccetera, eccetera.

Non staremo qui a ricordare, ancora una volta, il periodo di ferie tanto sospirato e che, finalmente, è direttamente "palpabile", a portata di mano.

Da alcune lettere giunte in Redazione risulta chiarissimo che molti vorrebbero partecipare, in qualche modo, a Commodore Computer Club ma, purtroppo, non tutti sono in grado di scrivere programmi sufficientemente interessanti, da proporre agli altri "soci" del Club.

Moltissimi lettori, però, non vivono di sola informatica e possono certamente "esprimersi" in altro modo. Di tanto in tanto, come avrete notato, pubblichiamo disegni, fotografie e, addirittura, fumetti inviati spontaneamente dagli utenti Commodore che, mettendo da parte listati e procedure, preferiscono evidenziare in senso... artistico la propria passione per il computer.

E allora, abbiamo detto, perchè non ufficializzare tale partecipazione offrendo a tutti i lettori la possibilità di veder pubblicati i propri lavori?

Ed ecco quindi l'idea, che è venuta approfittando del periodo particolarmente favorevole in cui esce questo stesso fascicolo:

Procuratevi carta e penna, oppure macchina fotografica e rullino, oppure "armate" il vostro computer con un programma di disegno o di digitalizzazione di immagini. Inviateci quindi i vostri capolavori tenendo conto delle nostre esigenze:

Disegni

Verranno presi in considerazione solo disegni (o brevi strip) eseguiti con inchiostro di *china*, su cartoncino bianco di formato massimo cm. 15 x 15.

Fotografie

Verranno pubblicate, se meritevoli, solo *diapositive* a colori, di qualsiasi formato.

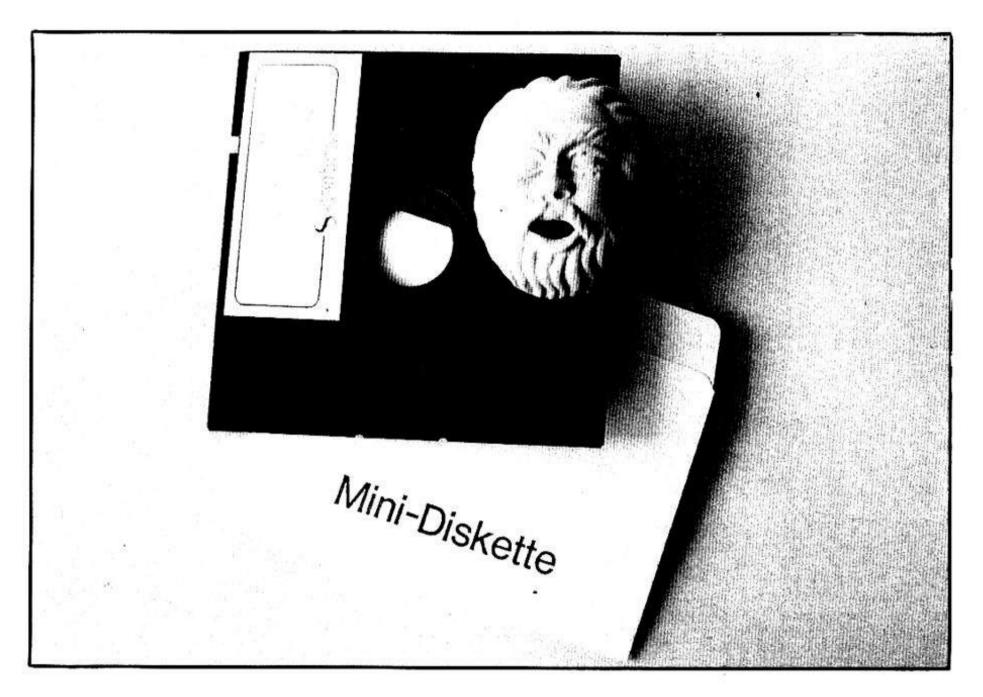
Elaborazioni computerizzate

Verranno esaminati esclusivamente *dischetti* contenenti le immagini digitalizzate e/o elaborate a mano successivamente. Evitate di inviare elaborazioni di immagini in movimento, in modo da consentirci di effettuare le fotografie della videata.

Nel caso di diapositive e digitalizzazioni computerizzate potrebbero comparire, nell'immagine stessa, alcune persone. In questo caso è indispensabile allegare, (debitamente firmata dall'interessato, sul retro di una fotocopia della sua carta di identità) l'autorizzazione alla pubblicazione dell'immagine in cui detta persona è ritratta.

Le immagini migliori verranno usate (sempre citando il loro autore) per illustrare le pagine di Commodore Computer Club.

Inviare il materiale a:

Commodore Computer Club
"Un'immagine per la rivista"
Via Mosè, 18
20090 OPERA (Mi)
Tel. 02/52.49.211

FLOPPY DISK, SE CI SEI BATTI UN BYTE

Prova a rispondere alle seguenti domande. La soluzione è in altra parte di questa stessa rivista

1/ I drive possiedono al loro interno un microprocessore?

☐ Sì, perchè altrimenti non sarebbero in grado di elaborare i vari dati.

☐ No: è il computer che, grazie ad opportune routine, attivate da specifici comandi, gestisce sia la parte elettromeccanica che elettronica.

2/ E' possibile, con opportuni programmi, "infettare" un dischetto con il 1541?

☐ Sì, perchè le tecniche di infezione da virus si possono applicare a qualsiasi computer dotato di drive.

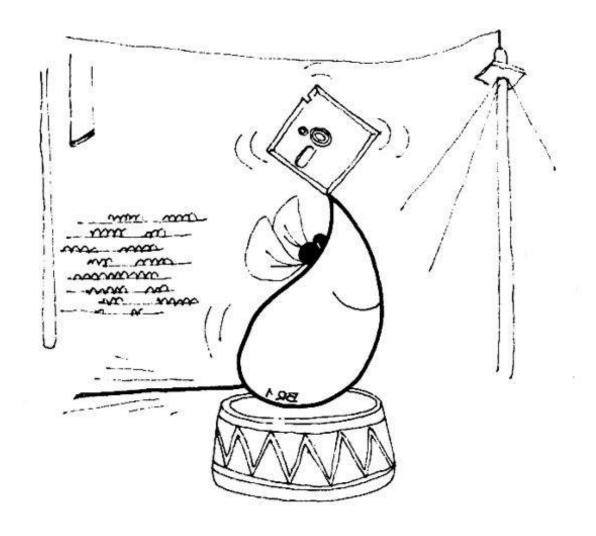
☐ No, perchè le infezioni da virus rappresentano un pericolo solo per l'Amiga ed i sistemi professionali in genere.

3/ Che cosa accade impartendo il comando...
Open 15, 8, 15, "I:Pippo"
Close 15

...nel tentativo di inizializzare il drive?

☐ Il led del drive lampeggia indicando un Syntax error.

☐ Il floppy viene inizializzato, ma solo se il suo nome è *Pippo*.

☐ Altro (specificare).

4/ Sul floppy è presente un file sequenziale di nome *Pippo*. Come gli si possono aggiungere altri caratteri, in coda?

☐ Non è possibile effettuare un'operazione del genere.

☐ E' possibile farlo solo se i caratteri da aggiungere sono alfa-numerici (e non grafici).

☐ Altro (specificare).

5/ Impartendo il comando...

Load "$:*=p", 8

.. si ottiene:

☐ Un Syntax error

☐ La ricerca di tutti i file il cui nome inizia con *P*

☐ Altro (specificare)

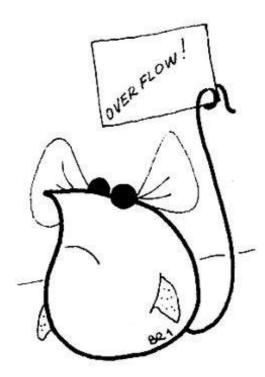
6/ Un floppy disk può essere usato, con un 1541, su entrambi i lati purchè:

☐ Si effettui una tacchetta, sul lato opposto, in corrispondenza a quella già presente.

☐ Si effettui un foro, sul lato opposto, in corrispondenza a quello già presente.

☐ Si effettuino entrambe le operazioni precedenti A, e B.

7/ E' possibile usare, durante una stessa elaborazione, due dischetti diversi, ma con *ID* identica?

☐ No, mai.

☐ Sì, purchè...

8/ Per errore abbiamo registrato un programma con *Save "Pippo, Poppo"* (cioè con una virgola nel nome). Che cosa è successo?

☐ Il led del drive ha lampeggiato durante la fase di registrazione: il programma non è stato registrato.

☐ Il programma è stato registrato, ma il suo nome è solo *Pippo* (cioè è stata omessa la parte successiva alla virgola).

☐ Il programma è stato registrato, ma il suo nome è *PippoPoppo* (cioè è stata omessa la virgola).

9/ Che cosa rappresentano, in un file di tipo programma, i primi due byte registrati sul floppy?

☐ La numerazione della prima riga Basic (forma basso / alto).

☐ Il tipo di file registrato, secondo un certo codice.

☐ La prima locazione in cui allocare il programma stesso (formato basso / alto).

10/ Che cosa accade cercando di caricare la Directory con *Load "$",8,1* e non con il solito Load "$",8?

☐ Nulla di particolare

☐ Altro (specificare).

11/ Che si intende con i termini *Allocare* e *Deallocare* riferiti al settore di un dischetto?

☐ Non sono altro che i sinonomi di *registrare* e *cancellare*.

☐ Rispettivamente, proteggere e sproteggere i dati registrati in quel settore.

☐ Non sono altro che i sinonomi di *registrare* e *caricare*.

12/ E' possibile scrivere qualche dato sul disco se la tacchetta di quest'ultimo è coperta?

☐ No, in nessun caso.

☐ Sì, purchè...

☐ No, purchè...

RISPOSTE A "FLOPPY DISK, SE CI SEI BATTI UN BYTE"

MICROPROCESSORI E DRIVE

Agli albori dell'informatica i microprocessori erano piuttosto costosi; di conseguenza la tendenza era quella di sfruttare tutte le loro risorse per svolgere i compiti più disparati, tra cui la gestione delle periferiche.

Con il passar del tempo, la drastica riduzione dei prezzi e, soprattutto, la necessità di aumentare la velocità di elaborazione, le case costruttrici decisero di dotare ciascuna periferica di un processore, incaricato di svolgere i vari compiti dopo aver ricevuto "ordini" opportuni dal calcolatore.

I primi drive dotati di microprocessore furono di marca Commodore, la quale seguì tale tradizione fino all'Amiga. Quest'ultimo computer, infatti, gestisce direttamente i drive che, quindi, sono costituiti da due motorini e da una manciata di circuiti minori. Il motivo dell'inversione di tendenza può essere attribuito alle notevoli potenzialità del microprocessore 68000 che non risentirebbe del maggior onere di lavoro; un altro motivo può esser costituito dalla notevole riduzione dei costi di produzione dei drive, che risultano più semplici da fabbricare in serie.

INFEZIONI

L'infezione di un dischetto, di solito, consiste nell'attivazione di un programma (di qualunque tipo essi siano) il cui nome inizia con "P".

1541 DOPPIA FACCIA

Per motivi di economia il drive 1541 viene prodotto con una sola testina di lettura - scrittura che agisce sulla parte inferiore del floppy inserito.

Per questo motivo un qualsiasi dischetto viene usato solo al 50% delle sue potenzialità nonostante entrambe le facce siano perfettamente magnetizzabili.

Effettuando, però, una tacchetta sul lato opposto, e girando ovviamente il disco in fase di inserimento, è possibile utilizzare la seconda faccia del floppy in maniera perfettamente identica a quella seguita per la faccia standard.

Facendo ruotare, a mano, il dischetto all'interno del suo involucro (trattandolo con la massima delicatezza) è possibile notare (sempre in linguaggio macchina) che tende a manifestare la sua presenza con la visualizzazione di messaggi più o meno fastidiosi per l'utente, oppure provocando danni a volte notevoli sui dischetti che vengono, successivamente, inseriti nel drive.

Un qualsiasi programma, pertanto, può contenere al suo interno una routine che, opportunamente inserita nel ciclo di interrupt, può provocare danni di vario genere.

Con il C/64, però, è relativamente semplice accorgersi della presenza di un virus e prendere opportuni provvedimenti. Se non bastasse, poi, i vari programmi commercializzati alterano quasi sempre il vettore di interrupt: se quindi è presente un virus, questo viene indirettamente "escluso" dai normali cicli di interruzione. Molto spesso, infine, è sufficiente la pressione di Run/Stop e Restore per ristabilire le condizioni standard operative.

Con l'Amiga, invece, la possibilità di multitasking rende più ardua l'individuazione di un eventuale virus, che può quindi diffondersi con imbarazzante rapidità.

Nel campo Ms-Dos quasi tutti i computer sono dotati di disco rigido, perennemente collegato con l'unità centrale. In questo caso l'azione di un virus, attivato anche una sola volta, può essere pericolosissima dal momento che, a seconda dell'infezione, si può arrivare alla formattazione completa dell'hard disk.

INIZIALIZZARE IL 1541

L'*inizializzazione* di un floppy per drive 1541 (operazione nettamente diversa da quella nota come *formattazione*) viene effettuata aprendo il canale 15 (dei comandi) ed inviando il carattere Ascii *I*. Qualunque carattere posto dopo questo carattere viene ignorato. Ne consegue che il comando indicato nella domanda ha lo stesso effetto di:

QUATTRO DOMANDE AI MEDIA D'AGENZIA E D'AZIENDA

1. Pianificate il mezzo "radio privata" o avete in programma di farlo? SI NO

2. Conoscete i palinsesti dei programmi di tutte le "radio private" per una precisa scelta del target group? SI NO

3. Avete la garanzia di aver ottenuto o di poter ottenere il miglior trattamento commerciale e creativo nella pianificazione di "radio private"? SI NO

4. Avete avuto la sicurezza e la garanzia ufficialmente certificata, da un istituto riconosciuto dalla categoria, della corretta avvenuta messa in onda dei comunicati pianificati sulle "radio private"? SI NO

- Se a queste 4 domande avete risposto sempre SI, avete già avuto rapporti con Egimedia e TIR Top Italia Radio.

- Se a queste 4 domande non avete risposto sempre SI, Egimedia e la AGB ITALIA, per TIR Top Italia Radio, sono l'**unico** interlocutore oggi che Vi farà rispondere SI a tutte le 4 domande ed altre ancora.

PER UN MIGLIORE SERVIZIO AI VOSTRI CLIENTI E ALLE VOSTRE AZIENDE

EGIMEDIA SRL - VIA DELLA SPIGA 1 - 20121 MILANO - TEL. 02/79.85.31 - 79.45.92

Open 15, 8, 15, "i"
Close 15

APPEND

L'operazione che consente di aggiungere caratteri in coda ad un file sequenziale già esistente è nota con il nome di *Append* ed è possibile con qualsiasi computer.

Nel caso del C/64, supponendo di avere su disco un file sequenziale di nome *Pippo*, è sufficiente...

Open 8,8,8, "Pippo,a"
Print# 8, "Caratteri da aggiungere"
Print #8, ...
Close 8

...per aggiungere tutti i caratteri (di qualsiasi tipo) che si desiderano. La particolare sintassi *",a"*, infatti, è specifica per l'operazione in oggetto che, lo ricordiamo, è impossibile da realizzare con un comune registratore a cassette.

SELEZIONE DI FILE

Un'opportunità offerta dal drive 1541, che sul manuale non viene purtroppo citata, è quella che consente di evidenziare solo i file di tipo ben definito, cioè Sequenziali, Programma, User e Relativo. Se, ad esempio, vogliamo conoscere i nomi di *tutti* i file di tipo *programma* presenti in un floppy, sarà sufficiente impostare...
Load "$:*=p", 8
...ed inserire, pertanto, l'iniziale del tipo di file (*P*, in questo caso). E' ovvio che al posto di P è possibile digitare una delle iniziali possibili (S, U, R). La sintassi, di solito, rimane valida anche usando velocizzatori sofware o hardware del tipo Speed Dos, Robcom, Final Cartridge, Niki e così via.

Il comando non deve esser confuso con la forma sintattica, piuttosto simile...
Load "$:p*", 8
...che, in effetti, individua i nomi di tutti i file che ad un determinato angolo di rotazione, compare un forellino in corrispondenza del piccolo foro dell'involucro.

Il motivo della presenza del forellino (posto in prossimità del foro centrale) è dovuta al fatto che alcuni computer, tra cui gli Ms-Dos compatibili, sfruttano (durante le normali operazioni del disco) il passaggio di un piccolo fascio di luce per individuare con precisione la "zona" del floppy su cui operano effettivamente.

Provando, infatti, ad "oscurare" detto forellino (con un po' di nastro adesivo) un computer Ms-Dos non lo "riconosce" più, nemmeno operando solo in lettura.

Il drive 1541, invece, opera solo avendo come riferimento zone magnetiche, alterabili via software. Il forellino, pertanto, è perfettamente inutile e non viene esaminato durante le normali operazioni (provate a coprire il forellino per verificare quanto affermato).

A PROPOSITO DI ID

Il codice ID, necessario per formattare la *prima* volta un dischetto, è di vitale importanza per la gestione del dischetto stesso.

Il sistema operativo del drive, infatti, ritiene di avere a che fare sempre con lo stesso dischetto anche se, durante una pausa di elaborazione, lo sostituiamo con un altro, dotato di ID identico.

Per ovviare a questo inconveniente è sufficiente, *prima* di effettuare operazioni con il drive, inizializzare il dischetto destinatario delle operazioni stesse.

In questo modo, infatti, è possibile usare più dischi dotati dello stesso ID. L'inizializzazione serve, anzi, proprio a questo: consentire al sistema operativo di "orientarsi" nel disco e di individuare il percorso ottimale sul quale "smaltire" le varie informazioni richieste dall'elaborazione.

UNO STRANO NOME

Il sistema operativo dei vecchi modelli di drive consentiva la possibilità di memorizzare file dotati di nomi contenenti anche caratteri illegali (doppio punto, virgola). I moderni drive, invece, "troncano" il nome all'ultimo carattere lecito.

Tentando, quindi, di registrare con...
Save "Pippo,Poppo", 8
...l'operazione si svolge senza intoppi ma, ad una successiva verifica sulla directory, ci accorgeremo che il nome, in realtà, è stato abbreviato in "Pippo".

PRIMI BYTE

Un file Sequenziale è formato esclusivamente dai byte che, in effetti, lo compongono, ad eccezione, ovviamente, di quelli che "legano" le varie parti del file sul dischetto stesso.

Un file programma, invece, è formato da tutti i byte che gli competono e da due byte (posti all'inizio del file) che rappresentano, nel formato alto - basso, l'esatta locazione in cui era presente il programma al momento della registrazione.

Per verificare quanto detto, provate a digitare il banalissimo programma...
10 rem prova
...con il nome...
Save "prova", 8
In seguito digitate il seguente listato:
10 Open 8, 8, 8, "prova, p"
20 get# 8, a$: Print asc(a$)
30 goto 20

Dopo il Run il computer visualizzerà i valori...
1, 8, 13, 8, 10
... prima di fermarsi con un "Illegal quantity error" (sul quale, tuttavia, non ci soffermiamo).

I primi due valori (1, 8) rappresentano la locazione di inizio del programma Basic registrato: 1 + 8 x 256 = 2049.

Gli altri due (13, 8) indicano, all'interprete Basic, la locazione in cui rintracciare la successiva linea Basic (che, nel nostro caso... non c'è!) Il numero 10, finalmente, indica il valore della prima (ed unica) riga Basic del programma in oggetto.

STRANA DIRECTORY

Per i motivi visti nella risposta precedente, i primi due byte della directory contengono valori che obbligano la visualizzazione del file a partire dalla prima locazione di schermo.

Alcune cartucce, invece, visualizzano sempre e comunque la directory nel modo consueto.

TOGLI E METTI

Quando si registra un qualsiasi file, ad esempio un programma, si occupa una porzione di spazio sul disco. Il sistema operativo, quindi, "deve" sapere che tale zona è occupata e che non può essere invasa da successive operazioni di scrittura.

In modo totalmente automatico, pertanto, il S.O. del 1541 annota le zone occupate (*allocandole* nella B.A.M, che è l'indice del dischetto) e le "libera" (le *dealloca*) nel caso l'utente le cancelli con il comando *scratch*.

Operando, invece, direttamente con le tracce ed i settori (servendosi dei comandi speciali del drive) non viene attivato l'automatismo. E' quindi compito dell'utente allocare (o deallocare) le zone del disco mediante gli idonei comandi *Block allocate* e *Block free*.

In questo modo è possibile, ad esempio, avere dischetti che presentano zero blocchi liberi ma che non visualizzano alcun nome nella directory!

PROTEZIONE

Per fortuna non è possibile scrivere (almeno sembra!) su un dischetto se la tacchetta di protezione è attivata. Questo particolare consente di proteggersi in modo adeguato da indesiderati attacchi dei nocivi virus del computer.

(IM)POSSIBILI RILOCAZIONI

I problemi della rilocazione di un programma scritto in Assembly, e la routine di individuazione di una cartuccia, sono argomenti complessi solo in apparenza

di **Domenico Pavone**

Dopo aver scritto un programma con l'editor del Macro Assembler (o con altri assemblatori, quali il *Merlin*), è necessario assemblarlo e salvarlo su disco.

In seguito, caricandolo in memoria con *Loader* (o funzioni analoghe), il programma funziona perfettamente grazie al comando di partenza, molto spesso una *Sys 49152*.

Se, però, si cerca di allocare lo stesso programma in altra area di memoria con l'opzione *Hex offset* del Loader, lo stesso sembra non reagire più alla nuova Sys di partenza (per esempio, 16834).

Il problema (sollevato da Raffaele Brusoni di Varese) è sicuramente da ricercarsi nella (non) rilocabilità del programma, che, tuttavia, con il Macro Assembler può essere ottenuta sempre, anche quando... non c'è.

Chiariamo meglio con un esempio pratico.

Si provi a digitare, con l'editor, il breve programma LM volutamente banale, mostrato in figura 1, che si limita a cambiare il colore del bordo dello schermo, ed a stampare un banale *OKAY*, con tecniche più volte affrontate (vedi anche l'inserto *Campus* n. 58 e 61) nell'ambito della rubrica riservata all'Assembly.

Poi lo si salvi con *Put"nomeprg"*, quindi, dopo essere usciti dall'editor (con *Kill*), si assembli la routine, assegnando (per esempio) il nome *obj* al file oggetto.

Infine, si carichi (Load .... 8) e si mandi in esecuzione (con Run) il "Loloader".

Premendo il solo Return alla richiesta "Hex offset?", si fornisca il nome del file oggetto (nel nostro caso "obj").

Quando il loader avrà concluso il suo compito, si digiti New, si provveda a caricare il "Monitor$8000" (con Load .... 8,1) e si impartisca Sys 32768.

Possiamo, a questo punto, esaminare il disassemblato della nostra miniroutine nei suoi effettivi indirizzi.

Il comando da utilizzare sarà...

D C000 C01D

...che visualizzerà quanto mostrato in figura 2, con i codici Ascii dei caratteri da stampare disposti da $C015 in poi.

Usciti dal monitor, si può constatare,

con Sys 49152, l'effettivo funzionamento del programma.

Fin qui niente di particolare, dato che, con la procedura seguita per attivare il loader, il programma è stato allocato a partire dall'indirizzo previsto nel programma sorgente di figura 1 ("*=$C000", ovvero: inizio da 49152).

Volendo installare la routine altrove, magari a partire da $4000 (decimale 16384), occorrerà rispondere *8000* alla richiesta di Hex Offset del Loader.

Così facendo, il valore $8000 (decimale 32768) verrà sommato all'indirizzo di inizio della routine, $C000 (49152).

E poichè, come dovrebbe esser noto, superato $FFFF (65535) il conteggio riprende da zero, in definitiva si otterrà un caricamento del codice macchina a partire da $4000.

Spento e riacceso il computer, si ripeta dunque la procedura di installazione in memoria del programma (con il Loader), seguendo le nuove indicazioni.

Dopodichè, si ricarichi il Monitor e, dopo l'ennesima Sys 32768, si digiti...

D 4000 401D.

La routine sarà presente nelle locazioni volute, ma, come mostrato in figura 3, alcune "incongruenze" ne impediranno un corretto funzionamento.

Più in dettaglio, l'istruzione 4006 è rimasta immodificata, e, andando a cercare i caratteri da stampare in $C015, non troverà proprio niente.

Per lo stesso motivo, il JMP $C006 (poco più in basso, 4011), non potrà che bloccare definitivamente la routine dal momento che è molto probabile che in quelle locazioni non sia presente un programma l.m. significativo (attenzione: se non avete spento e riacceso il computer potrebbe esser presente la routine di figura 2 e tutto *sembrerà* funzionare correttamente...).

Tutto in regola, invece, per il salto condizionato BEQ, che, come più volte ribadito, assicura una totale "trasportabilità" delle routine LM, pur se limitato ad un salto di soli 128 byte avanti e indietro.

Si potrà pensare, in definitiva, che la rilocabilità di una routine deve essere necessariamente prevista in fase di progettazione.

In effetti è proprio questa la soluzione migliore, ma... non è detta l'ultima parola.

Il Monitor del Macro Assembler, infatti, possiede anche un comando "N" (= New Locator), di una potenza straordinaria, anche se poco adoperato.

Consente, infatti, di rilocare tutti gli indirizzi assoluti di una routine, entro un "range" da noi voluto, semplicemente fornendogli un offset.

Più semplice a farsi che a dirsi, si digiti dunque...

N 4000 4014 8000 C000 D000

... e si riesamini il disassemblato con...

D 4000 4014

Se tutto è in regola, dovrebbe risultare uguale a quello mostrato in figura 4, con gli indirizzi (scorretti) prima visti in figura 3, ora tutti diretti al posto giusto (vedere, in particolare, le righe 4006 e 4011).

Un ritorno in ambiente basic, e relativa Sys 16384, confermerà il buon esito di tutte le nostre manovre.

I parametri di New Locator (N), vanno così interpretati:

```
FIGURA  1

001           *=$C000
002           LDA #$00
003           STA $D020
004           TAX
005 LOOP      LDA MSG,X
006           CMP #$0
007           BEQ FINE
008           JSR $FFD2
009           INX
010           JMP LOOP
011 FINE      RTS
012 MSG       .BYT 'OKAY',0
013           .END
```

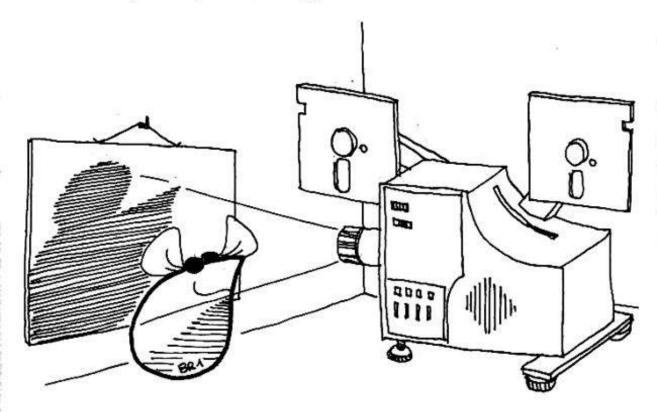
```
FIGURA  2

C000   LDA #$00
C002   STA $D020
C005   TAX
C006   LDA $C015,X
C009   CMP #$00
C00B   BEQ $C014
C00D   JSR $FFD2
C010   INX
C011   JMP $C006
C014   RTS
C015   ..........
```

```
FIGURA  3

4000   LDA #$00
4002   STA $D020
4005   TAX
4006   LDA $C015,X
4009   CMP #$00
400B   BEQ $4014
400D   JSR $FFD2
4010   INX
4011   JMP $C006
4014   RTS
4015   ..........
```

```
FIGURA  4

4000   LDA #$00
4002   STA $D020
4005   TAX
4006   LDA $4015,X
4009   CMP #$00
400B   BEQ $4014
400D   JSR $FFD2
4010   INX
4011   JMP $4006
4014   RTS
4015   ..........
```

4000 4014 = Indirizzi iniziale e finale della sezione da modificare.
.000 = Valore esadecimale (offset) da aggiungere agli indirizzi (valgono le stesse considerazioni fatte a proposito del Loader).
C000 D000 = Vengono modificati solo gli indirizzi compresi tra questi due valori; gli altri (tra cui $FFD2 e $D020 nella nostra routine) non subiscono variazioni.

Quanto visto, oltre che per l'uso dei loader, vale anche nel caso si operi un trasferimento con il comando T del monitor.

In pratica, grazie ad "N", è possibile rilocare qualunque routine, anche se normalmente "inamovibile".

IL C/64 SA LEGGERE?

In precedenti articoli abbiamo avuto modo di descrivere come intervenire sul Reset di sistema. Sollecitati da Aldo P. di Pordenone, ritorniamo sull'argomento ricordando che è necessario inserire gli indirizzi di una nostra routine l.m. nelle locazioni $8000 - $8001 e $8002 - $8003 e, subito dopo, memorizzare i caratteri "CBM80" in particolari locazioni di memoria.

Come fa il computer a capire se la stringa è proprio quella?

Più o meno, il procedimento è simile a quanto avviene per noi umani.

Quando, infatti, impariamo un nuovo vocabolo, in pratica lo memorizziamo da qualche parte nel nostro cervello e, ogni volta che lo sentiamo pronunciare (o lo leggiamo), non facciamo altro che paragonarlo alla struttura già "archiviata".

Se vi è una corresponsione, nel caso dell'uomo non necessariamente totale (c'è anche l'intuito, che diamine!) ecco che ne comprendiamo il significato.

Mettendo da parte ogni dissertazione psico-informatica, anche il nostro C/64 deve già avere in memoria un "termine di paragone" per poter giudicare se è in presenza di una corretta sequenza di caratteri o meno.

Nel caso in questione, possiamo verificarlo sperimentalmente con un monitor qualsiasi, anche se qui si farà riferimento alla sintassi del Macro Assembler.

Impartendo, infatti, un comando...
H D000 FFFF 'CBM80'
...si attiverà una ricerca dei caratteri racchiusi tra gli apici, dalla locazione $D000 fino alla fine della memoria, cioè $FFFF.

Si badi, però, che i caratteri CBM (e solo questi!) vanno digitati con lo Shift premuto, in quanto viene utilizzato dal sistema il loro codice Ascii + 128.

Dopo una breve elaborazione, verrà stampato sullo schermo un indirizzo: FD10.

Per dargli un'occhiata, basterà impartire, sempre da monitor, I FD10 (o, in altri tipi, M FD10).

La serie di codici carattere, si trova in fondo ad una Rom - Routine, quella appunto del Reset (per chi volesse disassemblarla, l'indirizzo iniziale è $FCE2), che, tra le altre cose, controlla appunto l'esistenza di eventuali Cartridge.

Il riconoscimento della stringa, avviene tramite un confronto, carattere per carattere, tra il contenuto delle locazioni da $8004 a $8008 e quello delle celle di memoria da $FD10 a $FD14.

Se non si verificano discrepanze, la presenza di una cartuccia esterna viene interpretata come certa.

"Lui", poverino, è sempre in buona fede...

E' IN EDICOLA VR VIDEOREGISTRARE

LA PRIMA RIVISTA DI VIDEOREGISTRAZIONE ATTIVA

UN VIDEOGAME PER L'ESTATE

Che cosa c'è di meglio di un joystick o un mouse se si hanno a disposizione computer del calibro di Amiga e C/64?

di **Andrea Ciaramella**

KICK OFF (Anco)
Amiga
(tra breve anche per C/64)

Vi ricordate della software house *Anco*? Divenne famosa grazie allo *Strip Poker*, ma per il resto ha sempre prodotto giochi mediocri e quasi del tutto sconosciuti al grande pubblico.

KICK OFF, al contrario, si distacchi dallo standard di basso livello e può esser definito, senza ombra di dubbio, il miglior gioco di calcio mai prodotto sino ad oggi per un personal computer. Solo *Microprose Soccer* (in versione C/64) riesce ad eguagliarlo.

Ma passiamo alla descrizione vera e propria del gioco: dopo il caricamento, lo schermo mostra un tabellone elettronico del tutto identico a quelli presenti, ormai, in quasi tutti gli stadi.

Nella fase iniziale (per mezzo del joystick) abbiamo la possibilità di selezionare un gran numero di opzioni tra cui: stabilire il numero di giocatori (due, oppure da soli contro il computer), se disputare un allenamento (in questo modo si ha tutto il campo a propria disposizione), giocare una sola partita oppure svolgere un intero campionato.

Attraverso le opzioni iniziali è persino possibile selezionare la disposizionedei giocatori in campo, secondo schemi classici.

Dopo le opzioni si passa al gioco vero e proprio; il campo è visto sempre dall'alto e sullo schermo è costantemente presente (nell'angolo in alto a sinistra) un "monitor" attraverso il quale è possibile esaminare la posizione contemporanea di tutti i giocatori, istante per istante.

Subitò dopo il fischio d'inizio ci si accorge della velocità impressionante del gioco che, soprattutto impostando il livello più elevato, lo pone un gradino più in alto di tutti gli altri giochi del genere.

In *Kick Off* sono presenti alcuni elementi che, ne siamo sicuri, lo renderanno in breve tempo il gioco più diffuso tra gli appassionati di calcio. Tra le novità introdotte sono da segnalare le dimensioni del campo (molto più grande di quelli offerti da giochi analoghi), la possibilità di commettere falli (con conseguenti ammonizioni, espulsioni o calci di rigore!) e, dulcis in fundo alla fine di ogni frazione di gioco si recupera il tempo perso per eventuali interruzioni dovute a falli.

La grafica non è certo un capolavoro ed anche il sonoro è ridotto ai soliti effetti (fischio dell'arbitro, urlo della folla in caso di goal, rumore dei calci inferti al pallone); la giocabilità di *Kick Off*, tuttavia, fa dimenticare questo (minimo) difetto.

La *Anco* ha di recente annunciato la versione, specifica per C/64, che, non appena la visioneremo, verrà presentata su queste stesse pagine.

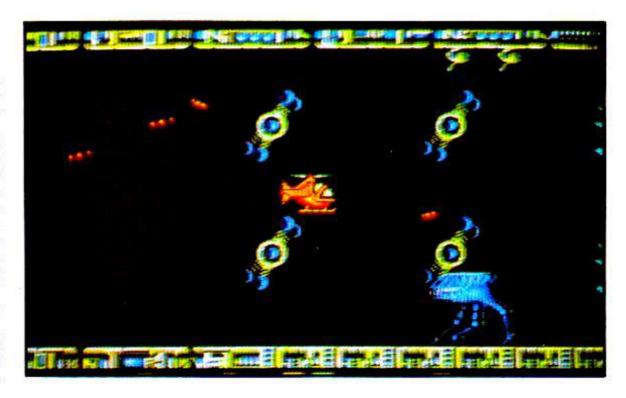
BLOOD MONEY (Psygnosis) *(Amiga)*

BLOOD MONEY è il seguito di *Menace*, il capostipite degli shoot'em up su Amiga: o, almeno, il primo tentativo apprezzabile.

Dopo una introduzione impressionante (durante la quale si ha la possibilità di ammirare asteroidi e pianeti che sembrano veri e di ascoltare voci sintetizzate davvero realistiche) si passa alla scelta del numero di giocatori; è possibile giocare in due o da soli e selezionare il pianeta da cui iniziare l'avventura. All'inizio, tuttavia, si può scegliere solo tra due dei quattro pianeti disponibili.

Terminata la fase iniziale veniamo catapultati nel gioco vero e proprio. Lo scopo, oltre a quello ormai classico di distruggere gli alieni, è quello di arricchirsi arraffando il denaro che rimane incustodito dopo la loro distruzione.

Il denaro servirà, poi, per comprare nuove armi nei "negozi" disseminati lungo l'intero percorso e per accedere agli altri mondi. Questi devono essere affrontati con astronavi più sofisticate, che necessitano di una quantità di denaro maggiore di quello che si ha inizialmente a disposizione.

Nel gioco si alternano fasi di scorrimento orizzontale e occasionali variazioni di direzione e livello (sopra e sotto la superficie). Queste ultime rappresentano una piccola innovazione nella consueta strategia di gioco.

Per quanto riguarda la grafica ed il sonoro, *Blood Money* è un vero e proprio capolavoro. Il livello sottomarino, ad esempio, è decisamente spettacolare.

Purtroppo, però, la giocabilità è in parte limitata dall'eccessiva difficoltà, raramente apprezzata dai videogiocatori non masochisti.

Speriamo che i programmatori di *Blood Money* si rendano conto della concreta abilità del giocatore "medio" e producano un terzo gioco che cosituirebbe, senz'altro, lo shoot'em up definitivo.

DRAGON'S LAIR (Readysoft) *(Amiga esp. 1 mbyte)*

Da alcuni anni, ormai, *Dragon's Lair* fa la parte del leone nelle sale giochi. E' stato il primo ad utilizzare il videodisco ed è diventato un vero e proprio "cult game" tanto che molti comprano Amiga, e relativa espansione, esclusivamente per avere quello che, a loro parere, è il miglior gioco per computer mai prodotto.

Passiamo alla trama: nei panni di *Dirk* (un guerriero senza macchia e senza paura) dovete penetrare nel castello di *Singe* e, passando attraverso 10 differenti livelli, liberare la principessa *Dafne* dalle grinfie del malvagio drago.

Il gioco (che trae origine, lo ricordiamo, da un flipper specifico equipaggiato con videodisco) utilizza ben 6 dischetti da 3 pollici e mezzo. La quantità di memoria utilizzata è quindi enorme: ma per che cosa viene utilizzata? Semplicemente per generare effetti grafici e sonori. Non esiste, infatti, un vero e proprio "gioco" perchè Amiga si limita ad animare il movimento giusto al momento giusto; ne consegue che imparando a memoria i movimenti il gioco è presto risolto.

Insomma, è vero che il gioco ha una grafica sensazionale ed un sonoro magnifico, ma l'abito non fa il monaco.

Il prezzo della confezione, poi, non aiuta la sua diffusione dal momento che si aggira attorno alle 100 mila lire.

MICROPROSE SOCCER (Microprose) *(Amiga - C/64)*

Era da tempo che aspettavamo l'adattamento per Amiga di *MICROPROSE SOCCER*, il gioco di calcio più famoso per il Commodore 64.

Vista la grandiosa giocabilità della versione ad 8 bit ci si aspettava, in verità, qualcosa di meglio.

Per coloro che non avessero mai visto *Microprose Soccer* su di un C/64 (credo che siano veramente pochi) riassumo, in breve, ciò che il gioco offre.

Tra le varie opzioni si può scegliere se disputare un campionato del mondo, un campionato tra le squadre dei nostri amici (possono partecipare, allo stesso campionato, sino a 32 persone diverse), una partita amichevole oppure il *Microprose Challenge*, che consiste nell'affrontare tutte le nazionali in ordine crescente di livello (l'Italia è la più forte, evviva l'Italia ehm... scusate, un eccesso di spirito patriottico).

Durante la partita è possibile effettuera colpi di testa, rovesciate e persino tiri ad effetto (denominati, nel gioco, *banana shots*) che spesso ingannano il portiere avversario.

Dopo la presentazione, si deve decidere se giocare all'aperto oppure in palestra. Nel caso scegliate la seconda opzione, non esistono falli laterali o calci d'angolo perchè la palla è respinta dai muretti posti ai bordi del campo (come nell'hockey).

Per rendere il tutto più realistico, i progettisti hanno deciso di inserire la pioggia, con conseguente scivolosità del campo (chiaramente solo giocando all'esterno) ed il replay con l'effetto moviola.

L'unico difetto di *Microprose Soccer* per Amiga è quello di non raggiungere il livello di qualità della versione per C/64, che lo supera perfino dal punto di vista grafico(!), e vi assicuro che è proprio così.

Se possedete un Amiga, e volete un gioco di calcio apprezzabile, non lasciatevi tentare da *Microprose Soccer*: potreste pentirvi di aver abbandonato il buon vecchio 64...

SILK WORM (Virgin games) *(Amiga - C/64)*

SILK WORM è uno dei pochi "spara e fuggi" di ambientazione militare presenti sul mercato; sicuramente il migliore.

Nel gioco è possibile mettersi al comando di un elicottero da guerra oppure di una jeep attrezzata con due cannoni, uno dei quali orientabile.

Lo scopo della missione è quello di distruggere tutti i nemici (in puro stile Rambo).

Credetemi, non sarà davvero facile portare a termine il compito; dovremo, infatti, combattere contro elicotteri avversari, postazioni lanciamissili, "mostri" d'acciaio che si materializzano davanti ai nostri occhi, mega carri armati ed altre diavolerie del genere.

Il tutto è, fortunatamente, reso più facile (pur restando molto difficile) dalla possibilità di giocare in due. Ed è proprio questa possibilità che aggiunge a *Silk Worm* una dose massiccia di divertimento. Vi potete divertire a scorrazzare pacificamente, magari insieme ad un amico, attraverso scenari disegnati in maniera superba e distruggere il nemico comune rimanendo illesi (o quasi).

Giocare a *Silkworm* significa non staccarsi più dal computer. Se, poi, vi scoprite guerrafondai (nel senso giocoso della parola) il videogame diventerà il vostro preferito in tempi brevissimi.

Entrambe le versioni sfruttano al massimo le capacità del computer su cui girano, ed anche nella versione ad 8-bit rimane inalterata la notevole giocabilità.

Gli effetti grafici e sonori sono magnifici e le varie "situazioni" che si verificano pongono il prodotto della *Virgin Games* nell'Olimpo dei videogiochi.

BLASTEROIDS (Image Works) *(C/64 - Amiga)*

Vi dice niente questo nome? Provate allora a togliere dal titolo i primi due caratteri, *BL*. Ciò che rimane è il nome di uno dei capostipiti dei videogiochi (quanto a popolarità viene subito dopo *Pong* e *Defenders*): è proprio ad *Asteroids* che il gioco si ispira. Il vostro compito, infatti, è quello di ripulire l'intera galassia da una improvvisa pioggia di asteroidi.

Il tutto è reso più sofisticato dalla possibilità di dotare la vostra astronave di nuovi poteri e di altre cose delle quali parleremo tra breve.

All'inizio della partita bisogna scegliere il settore dal quale partire per la missione; appena arrivati occorre sparare all'impazzata contro tutto e tutti: gli asteroidi che vagano per lo spazio, prima di essere distrutti, devono essere colpiti diverse volte ed ogni volta si dividono in parti più piccole fino a diventare polvere.

Una volta distrutti tutti gli asteroidi di un settore si passa a quello successivo, fino alla fine.

Durante il gioco appariranno, di tanto in tanto, astronavi nemiche che, dopo essere state distrutte (se, ovviamente, riuscite nell'intento...), sganceranno alcuni "moduli". Dovete raccoglierli per potenziare la vostra astronave in modo da viaggiare più velocemente, avere una potenza di fuoco maggiore, resistere meglio agli scontri con asteroidi, e così via.

Dopo aver ripulito tutti i settori ci si ritrova a scontrarsi, faccia a faccia, con *Mukor*. Egli, vera ed unica origine di tutti i vostri problemi, è un mostro deforme, munito di "protuberanze" dalle quali fuoriescono gli asteroidi che hanno invaso la galassia. Uccidete Mukor distruggendo le sue "protuberanze": l'universo intero ve ne sarà riconoscente.

Per quanto riguarda grafica e sonoro il gioco non brilla di certo, ma la giocabilità ed i poteri aggiunti lo rendono un titolo che non deve mancare agli appassionati di giochi spaziali.

SPEEDBALL
(Commodore 64)

Avete mai pensato a come potrebbe evolversi uno sport come il calcio negli anni futuri?

Allora avete pensato a *SPEEDBALL*, l'immaginario sport praticato intorno agli anni 2000 nel quale due squadre, formate da tre giocatori ciascuna, si affrontano all'interno di un campo d'acciaio lungo 160 piedi e largo 90.

Il campo è circondato da pareti alte 30 piedi che trattengono la palla (anch'essa d'acciaio, dal peso di 3 kg) che provocherebbe, in caso contrario, seri danni agli spettatori.

Lo scopo del gioco è quello di scagliare la palla nella porta avversaria (innovativo, vero?); altre regole vere e proprie non esistono.

All'inizio bisogna scegliere il numero di giocatori (al massimo due), se si vuole disputare una partita amichevole, oppure un campionato ad undici squadre.

Durante la partita appaiono in mezzo al campo dei simboli che offrono, a chi li raccoglie, interessanti possibilità come, ad esempio, fermare per dieci secondi i giocatori avversari o aumentare la *stamina* della vostra squadra (la *stamina* è la proprietà dei giocatori di essere veloci ed agili) oppure invertire il controllo del joystick dell'avversario e così via.

Altri simboli sono invece raccolti ed utilizzati come moneta per comprare goals oppure per comprare un minuto di tempo aggiuntivo ed altre azioni degne di ogni sport leale che si rispetti.

Il gioco è realizzato in maniera perfetta ed i suoi effetti grafici e sonori non hanno nulla da invidiare a quelli della versione Amiga (se poi si considera la differenza di prezzo, tutto diventa ancora più bello).

Se siete appassionati di calcio, questo potrebbe rappresentare una buona alternativa alle solite simulazioni; se non lo siete, provatelo lo stesso, non vi deluderà.

FORGOTTEN WORLDS
(Us gold)
(Amiga - C/64)

Avremmo potuto stupirvi con effetti speciali... e lo abbiamo fatto! Questo sembra essere il motto dei programmatori di *FORGOTTEN WORLDS*, un programma basato proprio sugli effetti speciali.

Il giocatore rappresenta un forzuto omaccione, armato con tanto di bazooka, al quale è stato affidato il compito di addentrarsi nei "mondi sconosciuti" (questa è la traduzione italiana del titolo), per controllare se sono abitati e, in caso affermativo, da chi.

Chiaramente i mondi sono abitati e, altrettanto chiaramente, i loro abitanti non sono certo fanciulle affascinanti, ma feroci alieni pronti a fare di voi un pasticcio di carne.

Ad aiutarvi nell'ardua impresa sono stati disposti dei "negozi" nei quali potrete comprare armi potenti ed armature sufficientemente resistenti.

Forgotten Worlds può essere giocato da uno o due giocatori contemporaneamente.

Il gioco trova il suo punto di forza, come dicevamo all'inizio, nella grafica e nel sonoro e molte volte vi capiterà di perdere la vita perchè distratti dagli scenari o dai mostri di fine livello (realmente terrificanti).

L'unico difetto del gioco è quello di offrire un sistema di gestione troppo difficile da governare, almeno per un principiante; in seguito, però, tutto diventa più facile: il gioco è certamente in grado di coinvolgervi completamente, ma non pensiamo che sia indispensabile alla vostra collezione; se però avete un gruzzolo in più, procuratevelo (il gioco, non il gruzzolo): ne vale la pena.

TIME SCANNER
(Activision)
(Amiga)

Ricordate la versione Coin-Op di *TIME SCANNER*? Fu un vero e proprio successo.

Oggi *Time Scanner* appare sui nostri monitor.

Avrà lo stesso successo? Proviamo ad esaminarlo insieme.

Si tratta di un gioco di flipper computerizzato ed il flipper, nella sua trasposizione su computer, non ha mai (o quasi) avuto lo stesso successo di un flipper vero e proprio.

Time Scanner aveva tutti i presupposti per diventare uno dei pochi giochi di flipper apprezzabili; tuttavia, pur rappresentando un adattamento di buon livello, presenta alcuni difetti che lo rendono meno divertente della "macchina elettronica" che possiamo trovare nei bar.

Il difetto maggiore che, per intenderci, "rovina" il gioco, è certamente quello del movimento della pallina, del tutto irregolare; nè la grafica (quasi uguale, in verità, a quella del Coin-Op) nè il sonoro riescono a risollevare più di tanto la spettacolarità... mancata del gioco.

Si ha la possibilità di viaggiare nel tempo. I vari periodi temporali sono rappresentati da quattro diversi campi di gioco, ognuno diviso in due schermi; è possibile viaggiare nel tempo facendo entrare la pallina nel *Time Tunnel*.

Il videogame riproduce piuttosto bene il gioco del flipper al punto che "scuotendo" il flipper, si rischia di modificare la traiettoria della pallina e si rischia perfino (qualche volta) il classico tilt.

Per gli appassionati di simulazioni di flipper *Time Scanner* non dovrebbe rappresentare un acquisto sbagliato; i patiti di altri generi di games, invece, farebbero bene a stargli alla larga.

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000
Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000
Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000
Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000
Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000
Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000
Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000
Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000
Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000
Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000
Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000
Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000
Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000
Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000
Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000
Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000
Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000
Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000
Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000
Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 28.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 615.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 8.930.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 9.180.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.965.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 2.300.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.715.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 3.050.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000
Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000
Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000
Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000
Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000
Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D
1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64
Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64
Permette il collegamento a Videotel, P.G.E. e banche dati - **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000
Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000
Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000
Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000
Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000
Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000
Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000
Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000
Nastro per stampante

Stampante MPS 1500C - L. 550.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000
Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

LOMBARDIA

Milano
- AL RISPARMIO - V.LE MONZA 204
- BCS - VIA MONTEGANI 11
- BRAHA A. - VIA PIER CAPPONI 5
- E.D.S. - C.SO PORTA TICINESE 4
- FAREF - VIA A. VOLTA 21
- FLOPPERIA - V.LE MONTENERO 31
- GBC - VIA CANTONI 7 - VIA PETRELLA 6
- GIGLIONI - V.LE LUIGI STURZO 45
- L'UFFICIO 2000 - VIA RIPAMONTI 213
- LOGITEK - VIA GOLGI 60
- LU - MEN - VIA SANTA MONICA 3
- MARCUCCI - VIA F.LLI BRONZETTI 37
- MELCHIONI - VIA P. COLLETTA 37
- MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
- NEWEL - VIA MAC MAHON 75
- PANCOMMERZ ITALIA - VIA PADOVA 1
- SUPERGAMES - VIA VITRUVIO 38
- 68000 E DINTORNI - VIA WASHINGTON 91

Provincia di Milano
- GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
- F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
- TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
- OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
- AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
- GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
- CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
- PENATI - VIA VERDI 28/30 - CORBETTA
- EPM SYSTEM - V.LE ITALIA 12 - CORSICO
- P.G. OSTELLARI - VIA MILANO 300 - DESIO
- CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
- COMPUTEAM - VIA VECELLIO 41 - LISSONE
- M.B.M. - C.SO ROMA 112 - LODI
- L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
- BIT 84 - VIA ITALIA 4 - MONZA
- IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
- I.C.O. - VIA DEI TIGLI 14 - OPERA
- R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
- ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
- TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
- NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
- COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
- ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
- IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE

Bergamo
- D.R.B. - VIA BORGO PALAZZO 65
- TINTORI ENRICO &C. - VIA BROSETA 1
- VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO

Provincia di Bergamo

BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
- COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
- B M R - VIA BUTTARO 4/T - DALMINE
- MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
- OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
- COMPUTER POINT - VIA LANTIERI 52 - SARNICO
- A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO

Brescia
- MASTER INFORMATICA - VIA F.LLI UGONI 10/B

PROVINCIA DI BRESCIA
- MISTER BIT - VIA MAZZINI 70 - BRENO
- CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
- VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
- MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
- BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
- INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
- "PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21

Como
- IL COMPUTER - VIA INDIPENDENZA 90
- 2M ELETTRONICA - VIA SACCO 3

Provincia di Como
- ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
- DATA FOUND - VIA A. VOLTA 4 -ERBA
- CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
- FUMAGALLI - VIA CAIROLI 48 -LECCO
- RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO

Cremona
- MONDO COMPUTER - VIA GIUSEPPINA 11/B
- PRISMA - VIA BUOSO DA DOVARA 8
- TELCO - P.ZZA MARCONI 2/A

Provincia di Cremona
- ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
- EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA

Mantova
- COMPUTER CANOSSA - GAL. FERRI 7
- 32 BIT - VIA C. BATTISTI 14
- ELET. di BASSO - V.LE RISORGIMENTO 69

Provincia di Mantova
- CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO

Pavia
- POLIWARE - C.SO C. ALBERTO 76
- SENNA GIANFRANCO - VIA CALCHI 5

Provincia di Pavia
- A. FERRARI - C.SO CAVOUR 57 - MORTARA
- LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
- M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO

Sondrio
- CIPOLLA MAURO - VIA TREMOGGE 25

Provincia di Sondrio
- FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO

Varese
- ELLE - EFFE - VIA GOLDONI 35
- IL C.TRO ELET. - VIA MORAZZONE 2
- SUPERGAMES - VIA CARROBBIO 13

Provincia di Varese
- BUSTO BIT - VIA GAVINANA 17-BUSTO A.
- MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
- PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
- GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
- J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE

Alessandria
- BIT MICRO - VIA MAZZINI 102
- SERV. INFOR. - VIA ALESSANDRO III 47

Provincia di Alessandria
- SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
- SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
- COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA

Asti
- ASTI GAMES - C.SO ALFIERI 26
- RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)

Cuneo
- ROSSI COMPUTERS - C.SO NIZZA 42

Provincia di Cuneo
- PUNTO BIT - C.SO LANGHE 26/C - ALBA
- BOSETTI - VIA ROMA 149 - FOSSANO
- COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO

Novara
- PROGRAMMA 3 - V.LE BUONARROTI 8
- PUNTO VIDEO - C.so RISORGIMENTO 39/B

Provincia di Novara
- COMPUTER - VIA MONTE ZEDA 4 -ARONA
- ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
- S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
- ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
- TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA

Torino
- ABA ELETTRONICA - VIA C. FOSSATI 5/P
- ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
- COMPUTER HOME - VIA SAN DONATO 46/D
- COMPUTING NEW - VIA M. POLO 40/E
- C.D.M. ELETTR. - VIA MAROCHETTI 17
- DE BUG - C.SO V. EMANUELE II 22
- DESME UNIVERSAL - VIA S.SECONDO 95
- FDS ALTERIO - VIA BORGARO 86/D
- IL COMPUTER - VIA N. FABRIZI 126
- MICRONTEL - C.SO D. degli ABRUZZI 28
- PLAY GAMES SHOP - VIA C. ALBERTO 39/E
- RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
- SMT ELETTRONICA - VIA BIBIANA 83/bis

Provincia di Torino
- PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
- BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
- HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
- MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
- I.C.S. - VIA TORINO 73 - IVREA
- DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
- EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
- DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
- FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
- GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE

Vercelli
- ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
- ELETTRONICA - STRADA TORINO 15

Provincia di Vercelli
- C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
- SIGEST - VIA BERTODANO 8 - BIELLA
- REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
- FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
- STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO

Belluno
- UP TO DATE - VIA V. VENETO 43

Provincia di Belluno
- GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE

Padova
- BIT SHOP - VIA CAIROLI 11
- COMPUMANIA - VIA T. CAMPOSANPIERO 37
- D.P.R. DE PRATO R. - V.LO LOMBARDO 4
- G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
- SARTO COMPUTER - VIA ARMISTIZIO 79

Provincia di Padova
- COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA

Treviso
- BIT 2000 - VIA BRANDOLINI D'ADDA 14
- GUERRA EGIDIO & C. - V.LE CAIROLI 95

Provincia di Treviso
- DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
- SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
- FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL

Venezia
- GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
- TELERADIO FUGA - SAN MARCO 3457

Provincia di Venezia
- GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
- REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE

Verona
- CASA DELLA RADIO - VIA CAIROLI 10
- TELESAT - VIA VASCO DE GAMA 8

• Provincia di Verona
- UBER - CP 0363(RAG.SOC. DERTA) - VIA MASCAGNI 31 -CASTEL D'AZZANO
- FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
- COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA

Vicenza
- ELET. BISELLO - V.LE TRIESTE 427/429
- SCALCHI MARKET - VIA CA' BALBI 139

Provincia di Vicenza
- SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
- GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA

Gorizia
- E.CO. ELETTRONICA - VIA F.LLI COSSAR 23

Trieste
- AVANZO GIACOMO - P.ZZA CAVANA 7
- COMPUTER SHOP - VIA P. RETI 6
- COMPUTIGI - VIA XX SETTEMBRE 51
- CTI - VIA PASCOLI 4

Udine
- MOFERT 2 - VIA LEOPARDI 21
- R.T. SISTEM UDINE - VIA L. DA VINCI 99

Provincia di Udine
- IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
- IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE

Bolzano
- COMPUTER POINT - VIA ROMA 82/A
- MATTEUCCI PRESTIGE - VIA MUSEO 54

Provincia di Bolzano
- RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
- ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
- ERICH KONTSCHIEDER - PORTICI 313 - MERANO
- ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO

Trent0
- CRONST - VIA G. GALILEI 25

Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA' 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL'UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA'DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G.FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF. - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R 2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 - PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090

SYSTEMS EDITORIALE PER TE

La voce III
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far *parlare* il computer, sia di farlo *Cantare!* Diversi esempi allegati.
Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.
Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.
Cassetta: L. 12000
Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).
Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese ed i guadagni di una famiglia.
Cassetta: L. 12000
Disco: L. 12000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.
Cassetta: L. 12000
Disco: L. 12000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.
Cassetta: L. 20000
Disco: L. 20000

Analisi di Bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.
Cassetta: L. 20000
Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 ed i rudimenti di programmazione. Interattivo.
Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.
Cassetta: L. 12000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore. Diversi esempi allegati.
Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.
Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.
Solo su disco: L. 25000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive. Ideale per studenti.
Disco: L. 25000

L.M. + Routine grafiche
Un fascicolo speciale (corredato di dischetto) suddiviso in due parti: corso completo di linguaggio macchina 6502 e implementazione di numerose routine che aggiungono al C/64 istruzioni Basic specifiche per la grafica, comprese quelle per disegnare in prospettiva!
Fascicolo + disco: L. 16000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.
Disco: L. 15000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.
Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.
Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su *"Commodore Computer Club"*. In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro. Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.
Ogni dischetto: L. 12.000

LIBRI TASCABILI

64 programmi per il C/64
Raccolta di programmi (giochi e utilità) semplici da digitare e da usare. Ideale per i principianti. ***(126 pag.)***
L. 4800

I miei amici C/16 e Plus/4
Il volumetto, di facile apprendimento, rappresenta un vero e proprio mini-corso di Basic per i due computer Commodore. Numerosi programmi, di immediata digitazione, completano la parte teorica. ***(127 pag.)***
L. 7000

62 programmi per C/16, Plus/4
Raccolta di numerosissimi programmi, molto brevi e semplici da digitare, per conoscere più a fondo il proprio elaboratore. Ideale per i principianti. ***(127 pag.)***
L. 6500

Micro Pascal 64
Descrizione accurata della sintassi usata dal linguaggio Pascal "classico". Completa il volume un programma di emulazione del PL/0 sia in formato Microsoft sia in vesione C/64 (da chiedere, a parte, su disco). ***(125 pag.)***

L. 7000

Dal Registratore al Drive
Esame accurato delle istruzioni relative alle due più popolari periferiche del C/64. Diversi programmi applicativi ed esempi d'uso. ***(94 Pag.)***

L. 7000

Il linguaggio Pascal
Esame approfondito della sintassi usata nel famoso compilatore. ***(112 pag.)***

L. 5000

Utilities e giochi didattici
Raccolta di numerosi programmi, in versione C/64 e Spectrum, di particolare interesse per chi intenda svliluppare software didattico. ***(127 pag.)***

L. 6500

Simulazioni e test per la didattica
Raccolta di numerosi programmi che approfondiscono e tendono a completare la trattazione già affrontata sul precedente volume. *(127 pag.)*

L. 7000

Dizionario del Personal Computer
Raccolta dei termini più diffusi nel campo professionale: dizionario inglese - italiano. (Edizione ridotta). ***(96 pag.)***

L. 8000

Dizionario dell'Informatica
Dizionario inglese italiano di tutti i termini usati nell'informatica. (Edizione completa). ***(385 pag.)***

L. 20000

Word processing: istruzioni per l'uso
Raccolta delle principali istruzioni dei più diffusi programmi di w/p per i sistemi Ms-Dos: Word Star, Samna, Multimate Advantage, Word 3. ***(79 pag.)***

L. 5000

Telefax
Volumetto divulgativo sull'importanza del Telefax e sulle sue modalità operative caratteristiche. ***(66 pag.)***

L. 5000

Come compilare un giornale aziendale in Azienda
I principali problemi per chi opera in ambiente DPT, affrontati e risolti con la massima chiarezza e semplicità. ***(80 pag.)***

L. 5000

Unix
Un volumetto per saperne di più sul sistema operativo professionale per eccellenza. ***(91 pag.)***

L. 5000

ABBONAMENTO

Commodore Computer Club

11 fascicoli: L. 50.000

ARRETRATI

Ciascun numero arretrato

di C.C.C. L. 5000

COME RICHIEDERE I PRODOTTI SYSTEMS

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, L. 3500 per spese di imballo e spedizione, oppure L. 6000 se si desidera la spedizione per mezzo raccomandata.
Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L. 50000.

Per gli ordini, compilare un normale modulo di C/C postale indirizzato a:

C/C Postale N. 37 95 22 07
Systems Editoriale
Viale Famagosta, 75
20142 MILANO

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche i prodotti desiderati ed il tipo di spedizione da effettuare.

Per sveltire la procedura di spedizione sarebbe opportuno inviare, a parte, una lettera riassuntiva dell'ordine effettuato, allegando una fotocopia della ricevuta del versamento.

Volendo una spedizione in contrassegno è necessario anticipare la cifra di L. 10000 (diecimila), da inviare secondo le modalità prima indicate, indipendentemente dalla quantità di materiale richiesto, e da conteggiare, comunque, IN AGGIUNTA alla cifra risultante dall'ordine. (Si sconsiglia, pertanto, la richiesta di di prodotti in contrassegno)

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare, oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a:

Systems Editoriale
Milano

N. 11 - Lire 12.000

Commodore 64 Club